



Modelo de bodega de datos para el manejo de variables matriciales

Andres Felipe Cadavid Agudelo

Universidad Nacional de Colombia
Facultad de Minas, Departamento de Ciencias de la Computación y de la Decisión
Medellín, Colombia
2019

Modelo de bodega de datos para el manejo de variables matriciales

Andres Felipe Cadavid Agudelo

Tesis o trabajo de investigación presentado como requisito parcial para optar al título de:
Magister en Ingeniería de sistemas

Director:

Ph.D Francisco Javier Moreno Arboleda

Línea de Investigación:

Bases de Datos

Universidad Nacional de Colombia

Facultad de Minas, Departamento de Ciencias de la Computación y de la Decisión

Medellín, Colombia

2019

Dedicatoria

*A Luis Fernando Cadavid Meneses, Luz Elena
Agudelo Toro, Abel Cadavid, Ligia Meneses y
Christian Camilo Cadavid Agudelo.*

Agradecimientos

Gracias a mis padres, a mis abuelos paternos y maternos por inculcar en mí el trabajo duro, la constancia y la necesidad insaciable de ser cada día mejor persona, gracias a familiares y amigos que constantemente motivaron cada etapa de mi vida, Gracias a mi hermano Christian Camilo Cadavid por ser mi mayor inspiración, por ser mi gran ejemplo a seguir, gracias al profesor Francisco Moreno por motivar mi carrera académica, lo considero de las personas más sabias que conozco.

Resumen

Los pictogramas (iconos) en los modelos conceptuales de bodegas de datos son elementos esenciales en el desarrollo de campos de estudio como la medicina, la geoinformática o la biología, estos aumentan la expresividad de los modelos y facilitan la construcción y el entendimiento de los mismos. Los pictogramas permiten representar estructuras complejas como trayectorias, puntos de interés, roles, entre otros. Por su parte, las aplicaciones espacio-temporales, promueven el uso de tipos de datos no convencionales como los objetos geométricos. Sin embargo, se observa una carencia de pictogramas para representar elementos del álgebra matricial como vectores y matrices, las cuales son ampliamente usadas en disciplinas como biología, análisis de imágenes o contabilidad. En este trabajo se presentan pictogramas que representan elementos propios del álgebra matricial como vectores, matrices u operaciones, lo que permite la generación de modelos conceptuales de bodegas de datos en áreas que usan álgebra matricial como insumo para la construcción de modelos matemáticos como la contabilidad matricial; allí, el uso de matrices, vectores y operaciones facilitan la consolidación de los estados financieros. La transformación de modelos conceptuales de bodegas de datos en modelos lógicos sobre sistemas de gestión de bodegas de datos facilita el planteamiento de consultas al incorporar funciones que operen tipos de datos complejos como arreglos de una dimensión (vectores) y de dos dimensiones (matrices), lo que enriquece a su vez el lenguaje SQL.

Palabras clave: Pictogramas, Modelos conceptuales, Bodega de datos, Modelos lógicos, Matriz, Vector, Contabilidad Matricial.

Abstract

Pictograms (icons) in conceptual models of data warehouses are essential elements in the development of fields of study such as medicine, geoinformatics or biology, these increase the expressiveness of models and facilitate the construction and understanding thereof. Pictograms allow for the depiction of complex structures such as trajectories, points of interest, roles, among others. For its part, the space temporary applications promotes the use of unconventional data types as Geometric objects. However, It is noticeable a lack of pictograms to represent matrix algebra elements like vectors and matrices, which are widely used in disciplines such as biology, image analysis or accounting. In this research paper are presented pictograms that represent elements of matrix algebra as vectors, matrices or operations, allowing the generation of conceptual models of data warehouses in areas that use matrix algebra as input for the construction of mathematical models such as matrix accounting, There, the use of matrices, vectors and operations facilitate the consolidation of financial statements. Transforming conceptual models of data warehouses into logical models over database management systems facilitates the queries proposal by incorporating functions that operate complex data types such as one-dimensional arrays (vectors) and two-dimensional arrays (matrices), enriching in turn the SQL language. Which enriches in turn the SQL language.

Keywords: Pictograms, conceptual models, data warehouse, logical models, vector, matrix accounting.

Contenido

	Pág.
Resumen	IX
Lista de figuras.....	XIII
Lista de tablas	1
Introducción	2
1. Enriquecimientos pictóricos y conceptuales sobre modelos de bodegas de datos	5
1.1 Conceptos de BD	6
1.1.1 Modelos conceptuales de BD	6
1.1.2 Esquema en estrella	6
1.1.3 Esquema en copo de nieve	8
1.2 Modelos conceptuales enriquecidos para BD	9
1.3 Comparación y clasificación de modelos para BD	20
1.4 Área de estudio por explorar	22
2. Matrices aplicadas	23
2.1 Matriz de Leslie	23
2.2 Matrices en imágenes digitales	26
2.3 Contabilidad Matricial	26
2.3.1 Contabilidad clásica (contrapartida)	27
2.3.2 Ejemplo Contrapartida	28
2.3.3 Contabilidad Matricial.....	29
2.4 Bases de datos matriciales.....	31
3. Bodegas de datos con objetos matriciales	35
3.1 Bodega de datos convencional hechos contables	35
3.2 Ejemplo movimientos de hechos contables	37
3.3 Agregaciones pictóricas y modelo de bodegas de datos enriquecido	40
4. Implementación modelo bodega de datos con variables matriciales para el manejo de hechos contables.....	50
4.1 Entorno de desarrollo	50
4.2 Implementación de modelos físicos a partir de modelos conceptuales.....	51
4.2.1 Funciones Definidas por el usuario para el manejo de arreglos	53
4.3 Consultas Modelo lógico convencional y matricial	54
4.3.1 Consulta 1	54

4.3.2 Consulta 2	55
4.3.3 Consulta 3	57
5. Conclusiones y recomendaciones	59
5.1 Conclusiones	59
5.2 Recomendaciones	61
A. Anexo: Tablas modelo lógico	63
A.1 Tablas de dimensiones	63
A.2 Tablas de Hechos y vista	64
B. Anexo: Funciones definidas por el usuario Hive-Java.....	67
B.1 Función MAT_SUM_ALL.....	67
B.2 Función MAT_SUB_VEC	69
B.3 Función MAT_SUM_COL.....	70
B.4 Función MAT_SUM_ROW	71
Bibliografía	73

Lista de figuras

	Pág.
Figura 1-1: Esquema en estrella.....	7
Figura 1-2: Dimensión Producto. Modificado a partir de (Kimball & Ross, 2013).	8
Figura 1-3: Esquema en copo de nieve.	8
Figura 1-4: Dimensión Producto, esquema en copo de nieve. Modificado a partir de (Kimball & Ross, 2013).....	9
Figura 1-5: Iconos espaciales: (a) tipos de datos y (b) relaciones topológicas (Wagner et al., 2014).	10
Figura 1-6: Modelo multidimensional para el análisis de mantenimiento de carreteras (Highway maintenance) (Malinowski & Zimanyi, 2005).	10
Figura 1-7: BD de episodios (Episodes)(Campora et al., 2011).	11
Figura 1-8: Estereotipos e iconos usados en el esquema de una BD (Oueslati & Akaichi, 2014).	12
Figura 1-9: Fragmento de un modelo conceptual para una BD de trayectorias de ambulancias (Oueslati & Akaichi, 2014).	13
Figura 1-10: Extensiones para la representación semántica de trayectorias (Jenhani & Akaichi, 2014).	13
Figura 1-11: Representación en niveles de abstracción de una trayectoria (Jenhani & Akaichi, 2014).	14
Figura 1-12: Modelo conceptual para una BD de electrocardiogramas (Arigon et al., 2006).	15
Figura 1-13: Modelo conceptual para una BD de bioinformática (Arigon et al., 2006).	15
Figura 1-14: Modelo conceptual para una BD de trayectorias con 5W1H (Wagner et al., 2014).	16
Figura 1-15: Disyunción exclusiva en relaciones espaciales (Moreno et al., 2010).....	16
Figura 1-16: Tipos de inclusión y disyunción: a) inclusión total y parcial, b) ejemplo de inclusión parcial y c) ejemplo de inclusión total (Moreno et al., 2010).	17
Figura 1-17: Relación temporal y geometría variable en el tiempo (Moreno et al., 2010).	17
Figura 1-18: Representación de elementos de una BD con elementos del modelo ER: a) hecho, b) niveles-entidades y c) Relaciones entre entidades (Tryfona et al., 2004).	18
Figura 1-19: Relaciones entre entidades: (a) representación y (b) ejemplos (Tryfona et al., 2004).	18
Figura 1-20: Hechos, dimensiones y jerarquías StarER (Tryfona et al., 2004).	19
Figura 1-21: Tipos de atributos de hechos y representación gráfica(Tryfona et al., 2004).	19

Figura 1-22: Modelo Espacio – temporal (Zimányi, 2012).	20
--	----

Figura 2-1: Matriz de Leslie (Bock et al., 2016).	25
Figura 2-2: Ejemplo de cuenta T para la cuenta de activo Banco.	27
Figura 2-3: Cuentas T- Estado inicial.	28
Figura 2-4: Cuentas T - Estado final.	29
Figura 2-5: Matriz de cuentas (Poca Freddy, 2017).	29
Figura 2-6: Consulta distancia.	32
Figura 2-7: Consulta distancia optimizada.	32
Figura 2-8: Resultados cálculo de la matriz de Gram.	34
Figura 2-9: Resultados regresión Lineal.	34
Figura 2-10: Resultados detección de Outlier.	34

Figura 3- 1: Notación de Malinowski y de Zimanyi (Malinowski & Zimányi, 2008): a) dimensión con jerarquía básica (un solo nivel), b) dimensión con jerarquía compuesta por dos niveles, c) cardinalidades, d) criterio de análisis y e) hechos y medidas (measures).	35
Figura 3- 2: Modelo conceptual para una BD de movimientos de hechos contables.	36
Figura 3-3: Dimensión Cuenta y sus niveles: a) Tipo Cuenta y b) Cuenta.	38
Figura 3-4: Dimensión Cliente y su nivel Cliente.	38
Figura 3- 5: Dimensión Hecho Contable y sus niveles: a) Tipo Hecho Contable y b) Hecho Contable.	39
Figura 3-6: Tabla de hechos (movimiento contable).	40
Figura 3-7: Modelo Conceptual BD hechos contables en forma matricial.	43
Figura 3- 8: Tabla de Hechos con elementos del álgebra matricial.	44
Figura 3- 9: Ejemplo de generación de los vectores virtuales Débitos y Créditos.	45
Figura 3-10: Matriz de cuentas Matriz_Saldo .	45
Figura 3-11: Obtener el saldo de las cuentas que tuvieron movimientos el 4 y 6 de mayo de 2018 BD convencional.	47
Figura 3-12: Obtener el saldo de las cuentas que tuvieron movimientos el 4 y 6 de mayo de 2018 BD Matricial.	48
Figura 3- 13: Iteración sobre el Hecho.	49
Figura 3-14: Tabla de hechos agrupados por Mes.	49

Lista de tablas

	Pág.
Tabla 1-1: Comparación y clasificación de modelos para BD.....	20
Tabla 2-1: Parámetros demográficos de iguanas verdes hembras del nido Slothia – Panamá (Bock et al., 2016).	24
Tabla 2-2: Tipos de cuentas (Lucía Urueña & San Mateo, 2010).	27
Tabla 2-3: Ejemplo de matriz de cuentas.	30
Tabla 2-4: Consolidación: vectores de créditos y de débitos.	30
Tabla 2-5: Vectores de débitos, de créditos y de saldos finales.	31
Tabla 3-1: Tipos de datos (objetos) del álgebra matricial para el modelo conceptual de una BD.	41
Tabla 3-2: Operaciones del álgebra matricial.	41
Tabla 3-3: Operaciones entre objetos.	44
Tabla 3-4: Comparación de consultas en la BD convencional y en la matricial.	46
Tabla 4-1: Estructura tabla de hechos modelo convencional.	51
Tabla 4-2: Estructura tabla de hechos modelo matricial.	52
Tabla 4-3: Funciones definidas por el usuario.	53
Tabla 4-4: Tiempo ejecución consultas.	58

Introducción

Los enriquecimientos pictóricos y conceptuales sobre modelos convencionales de bodegas de datos (BD) ¹ han aumentado en los últimos años. Uno de los objetivos es formalizar el manejo de tipos de datos complejos (i.e., datos no escalares) en áreas del conocimiento como la medicina, la bioinformática y la geo informática entre otras (Arigon, Tchounikine, & Miquel, 2006). A raíz de esto se han propuesto BD espaciales y BD espacio-temporales (Malinowski & Zimanyi, 2007), donde el componente espacial o espacio-temporal complementa a los datos convencionales y permite análisis más detallados. Para ello, en los modelos de BD conceptuales convencionales (esquema en estrella y copo de nieve) (Silvers, 2008) se suelen incluir elementos pictóricos que representan tipos de datos geométricos. Con base en lo anterior surgen campos de estudio, como las BD de trayectorias, donde el principal objetivo es describir el comportamiento de objetos en movimiento, definir sus desplazamientos y paradas (visitas) en sitios interesantes para la aplicación (e.g., un turista visita un museo o un restaurante); los sitios que son visitados por los objetos móviles son denominados puntos de interés y las rutas hechas por los objetos trayectorias. El análisis de estas últimas tiene como objetivo principal comprender el comportamiento del movimiento de los objetos. Como ejemplo considérese el desplazamiento de las aves en migración o la ruta trazada por un conductor en una ciudad. La captura de estos datos se facilita gracias a los avances tecnológicos que se han incrementado en la última década (Wagner et al., 2014) donde los dispositivos móviles cuentan con sistemas de geolocalización, los cuales recopilan y frecuencia establecida (e.g., cada n segundos). Una trayectoria se puede considerar como una secuencia o

¹ BD es una sigla que se suele usar para base de datos; sin embargo, en este documento siempre se usará como una sigla para bodega de datos.

arreglo de puntos (coordenadas con latitud, longitud y tiempo (D. Orlando, Frade, Nelson, & Castillo, 2007).

Los enriquecimientos pictóricos sobre BD carecen de elementos que describan componentes de álgebra matricial como vectores o matrices, usados en áreas de estudio como la biología o contabilidad matricial (Goicoechea et al., 1991); por lo tanto, se proponen pictogramas para representar elementos propios del álgebra matricial como vectores, matrices y operaciones, lo que permite la generación de modelos conceptuales de BD en áreas que usen álgebra matricial como insumo para la construcción de modelos matemáticos como la contabilidad matricial, deben considerarse también los tipos de operaciones permitidas entre los objetos del álgebra matricial (Gamboa & Rodríguez, 2003), ya que no todo tipo de operación entre estos es correcta o tiene sentido, allí las restricciones juegan un papel fundamental para determinar qué tipo de objetos pueden ser agregados en un modelo particular. Los modelos construidos con las adiciones propuestas permiten describir con mayor precisión los dominios referenciados y a su vez facilitan el planteamiento de modelos lógicos sobre sistemas de gestión de BD como Hive o Teradata, donde se deben desarrollar funciones definidas por el usuario para manejar y operar tipos de datos complejos como arreglos de 1 y 2 dimensiones, ya que estos sistemas de gestión de BD no cuentan con todos los tipos de operaciones necesarias para resolver las preguntas particulares de cada dominio, el uso de funciones definidas por el usuario permite el planteamiento de consultas intuitivas y con mejor rendimiento que consultas planteadas sobre modelos lógicos convencionales que simulen un comportamiento matricial, además las nuevas funciones propuestas o desarrolladas por el usuario enriquecen las sentencias SQL disponibles por los diferentes sistemas de gestión de BD. La incorporación de objetos matriciales en los modelos conceptuales de BD permite mejorar áreas de estudio como la contabilidad matricial, donde las cuentas T de un estado financiero son volcadas en matrices y de esta manera es más fácil la consolidación de los estados financieros al final de periodos establecidos.

El documento se organiza de la siguiente manera: en el capítulo uno se muestran los diferentes enriquecimientos o mejoras propuestos por diversos autores a modelos conceptuales de BD; el capítulo dos muestra el uso de componentes del álgebra matricial en diversas áreas de estudio como la biología, el análisis de imágenes y la contabilidad; además, se muestra una introducción del manejo de escalares, vectores y matrices en las

bases de datos; el capítulo tres introduce los pictogramas para el manejo de objetos pertenecientes al álgebra matricial como matrices, vectores y operaciones dentro de modelos conceptuales de BD y se desarrolla un modelo conceptual para el manejo de hechos contables con las adiciones propuestas; en el capítulo cuatro se implementa el modelo conceptual de BD para el manejo de hechos contables en un modelo lógico de BD sobre Hive, incluyendo en este sistema funciones definidas por el usuario para el manejo de variables tipo arreglo de 1 y 2 dimensiones y en el capítulo cinco se presentan las conclusiones de la investigación.

1. Enriquecimientos pictóricos y conceptuales sobre modelos de bodegas de datos

Los modelos conceptuales convencionales para bodegas de datos(BD) (Gosain & Singh, 2014) carecen de elementos pictóricos (iconos) especializados que faciliten la representación de aspectos inherentes de dominios como la medicina, la bioinformática y la geoinformática, entre otras (Arigon et al., 2006).

En particular, los avances tecnológicos de la última década han impulsado la geoinformática(Wagner et al., 2014). Por ejemplo, los dispositivos móviles suelen disponer de sistemas de geolocalización, los cuales generan datos sobre la ubicación de un objeto (una persona, un animal, un vehículo) con una determinada frecuencia (e.g., cada cinco segundos). Los modelos conceptuales enriquecidos para BD en dominios como la geoinformática(Malinowski & Zimanyi, 2005) suelen incluir iconos como puntos, líneas o polígonos que representan objetos del mundo real tales como el sitio donde está ubicado un semáforo (latitud y longitud), una vía o la región que delimita un barrio.

Por otro lado, las BD *espacio-temporales* propician el análisis del desplazamiento de objetos móviles, i.e., el análisis de sus trayectorias(Malinowski & Zimanyi, 2007). Una trayectoria se puede representar mediante una secuencia de puntos (latitud, longitud, altitud y tiempo) ordenados ascendentemente según el tiempo (S. Orlando, Orsini, Raffaeta, Roncato, & Silvestri, 2012). El análisis de trayectorias suele incluir aspectos como la velocidad, los sitios visitados por el objeto móvil, la distancia recorrida, la orientación (norte, sur), entre otros. Por ejemplo, en Jenhani & Akaichi(2014) se presenta una aplicación médica para el análisis de trayectorias de ambulancias. Los autores establecen como sitios de interés aquellos donde se detienen las ambulancias, e.g., el sitio donde se recoge un herido o el hospital donde este es llevado; además se analiza la velocidad promedio de las ambulancias y las vías que suelen usar para sus recorridos. Por su parte, en Oueslati & Akaichi (2012) se propone un modelo conceptual enriquecido para una BD

con iconos para representar los diferentes tipos de dimensiones para el análisis de trayectorias de ambulancias, i.e., cada dimensión se representa con un icono distintivo, e.g., la dimensión temporal se representa con un plano cartesiano y un reloj mientras que la dimensión espacial (geográfica) se representa con un plano cartesiano y un polígono.

En este documento se analizan diferentes modelos conceptuales enriquecidos (con iconos) para BD. El objetivo es doble:

- Presentar un panorama actual sobre el tipo de extensiones (enriquecimientos) que han sido propuestas, su utilidad y el dominio al cual están dirigidos.
- Identificar áreas o problemas para los cuales no existen modelos conceptuales de BD especializados, i.e., carecen de iconos deseables para la representación de determinados conceptos.

1.1 Conceptos de BD

Inmon (2002) y Kimball (2013) son considerados los padres de las BD. Una BD es una base de datos especializada para soportar la toma de decisiones y usualmente se modela mediante una visión multidimensional de los datos (Moreno, Echeverri, & Arango, 2010).

Las fuentes de datos de una BD suelen ser bases de datos transaccionales (Reinosa, 2012). Los datos que van a ser almacenados en la BD se someten a un proceso conocido como ETL (extract, transform, and load), i.e., extracción, transformación y carga (Sillers, 2008). El objetivo de este proceso es estandarizar y estructurar, en la BD, los datos provenientes de las diferentes fuentes de datos.

1.1.1 Modelos conceptuales de BD

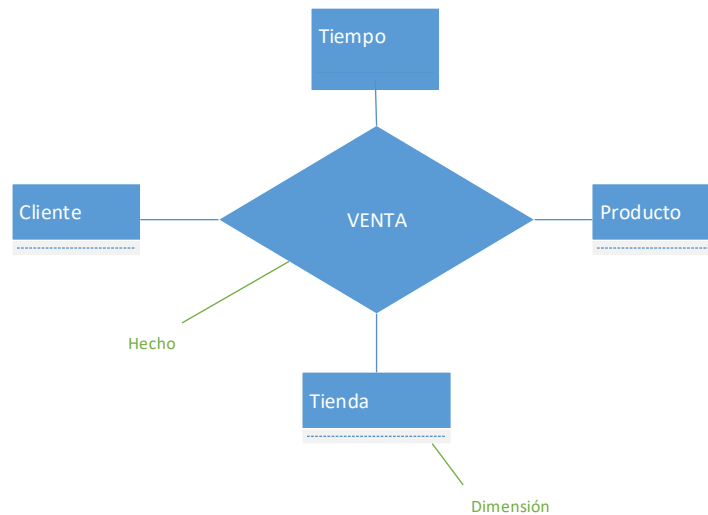
Los modelos conceptuales convencionales para BD (Kimball & Ross, 2013) suelen representar los hechos y las dimensiones mediante dos esquemas, estrella (star) y copo de nieve (snowflake).

1.1.2 Esquema en estrella

En este esquema, los datos de cada dimensión se almacenan en una sola tabla. Es decir, desde el punto de vista de una base de datos relacional, la tabla que almacena los datos

de una dimensión suele tener un grado de normalización bajo (e.g., primera o segunda forma normal). Un ejemplo se muestra en la Figura 1-1: Esquema en estrella. Allí se tiene el ejemplo típico de una BD con el hecho Ventas y las dimensiones Cliente, Tiempo, Tienda y Producto.

Figura 1-1: Esquema en estrella.



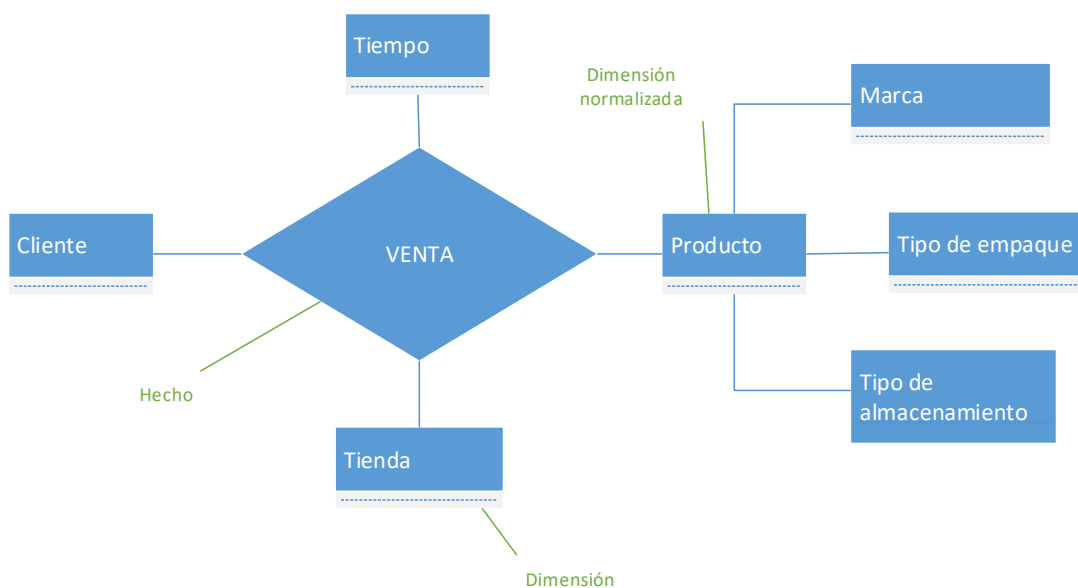
Por ejemplo, la dimensión Producto se almacena en una tabla con la estructura mostrada en la Figura 1-2.

Figura 1-2: Dimensión Producto. Modificado a partir de (Kimball & Ross, 2013).

Producto
Código producto (PK)
Número producto
Descripción producto
Tipo marca
Categoría
Departamento
Tipo de empaque
Indicador abrasivo
Tamaño
Tamaño por unidad
Tipo de almacenamiento
Tipo ciclo de vida
Ancho estante
Alto estante
Profundidad estante
...

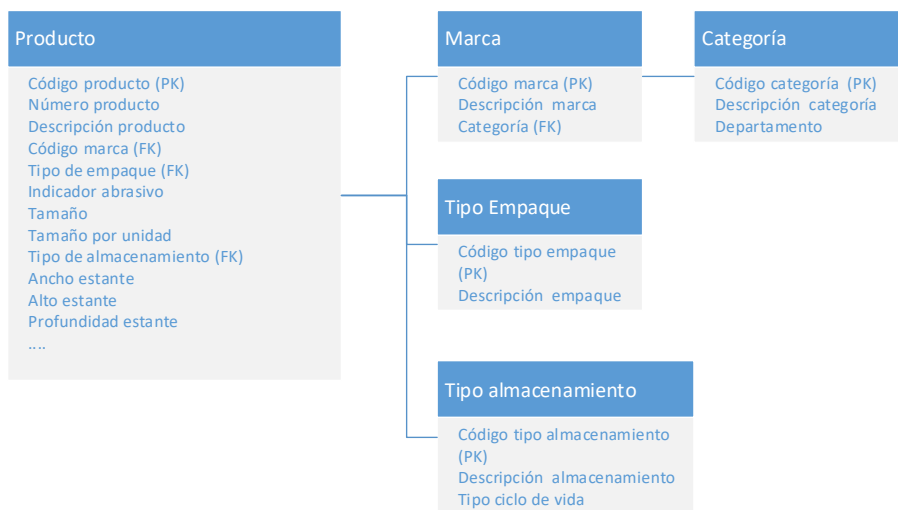
1.1.3 Esquema en copo de nieve

En este esquema se normalizan las dimensiones. De esta forma, una dimensión se implementa usualmente mediante varias tablas (una para cada nivel de la dimensión) (Kimball & Ross, 2013). Un ejemplo se muestra en la Figura 1-3.

Figura 1-3: Esquema en copo de nieve.

Aquí, a diferencia de la dimensión Producto de la Figura 1-1, los niveles que componen a esta dimensión se hacen explícitos y cada uno requerirá una tabla dentro de un modelo lógico. La Figura 1-4 detalla la estructura de la dimensión Producto (normalizada) en un esquema en copo de nieve.

Figura 1-4: Dimensión Producto, esquema en copo de nieve. Modificado a partir de (Kimball & Ross, 2013).

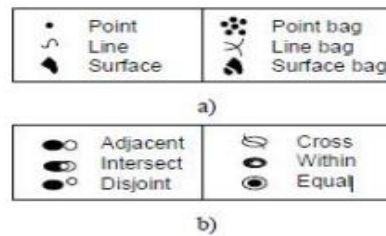


1.2 Modelos conceptuales enriquecidos para BD

Como se expresó en la introducción, los modelos conceptuales convencionales para BD carecen de iconos especializados que faciliten la representación de aspectos de determinados dominios. Por ejemplo, se ha estimado que un 80% de los datos almacenados en una BD contienen componentes espaciales (Malinowski & Zimanyi, 2007). Esto ha impulsado el enriquecimiento de los modelos conceptuales para BD con iconos para representar este tipo de datos.

Malinowski y Zimányi (2005) incorporan iconos para representar tipos de datos espaciales y relaciones topológicas a un modelo conceptual multidimensional con medidas (las medidas hacen parte de los hechos) y dimensiones espaciales. Los iconos son tomados de (Parent, Spaccapietra, & Zimányi, 2004), ver Figura 1-5.

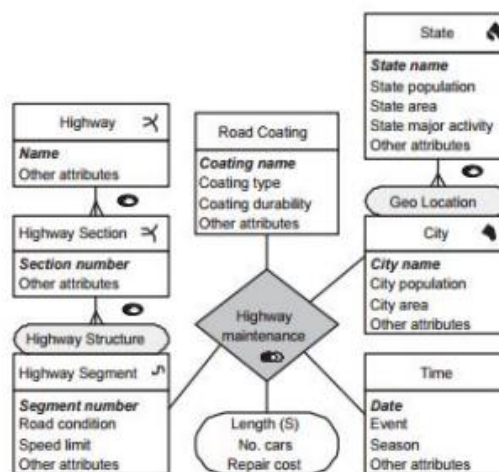
Figura 1-5: Iconos espaciales (a) tipos de datos y (b) relaciones topológicas (Wagner et al., 2014).



La Figura 1-5a muestra los iconos (tipos de datos espaciales) para representar objetos espaciales. Estos son un punto, una línea, una superficie (región, polígono), un conjunto de puntos, de líneas y de superficies.

La Figura 1-5b muestra los iconos para representar las relaciones topológicas entre los objetos espaciales. Estas son adyacencia (adjacent), intersección (intersect), disyunción (disjoint), cruce (cross), inclusión (within) e igualdad (equal). Por ejemplo, sean dos niveles City (Ciudad, representada con una superficie) y State (Estado, representado con un conjunto de superficies) de una dimensión geográfica. Entre estos niveles existe una relación espacial (within) que indica que una ciudad (superficie) está incluida (dentro, within) de un estado (conjunto de superficies). En la Figura 1-6 se muestra un ejemplo del modelo de Malinowski y Zimányi (2005).

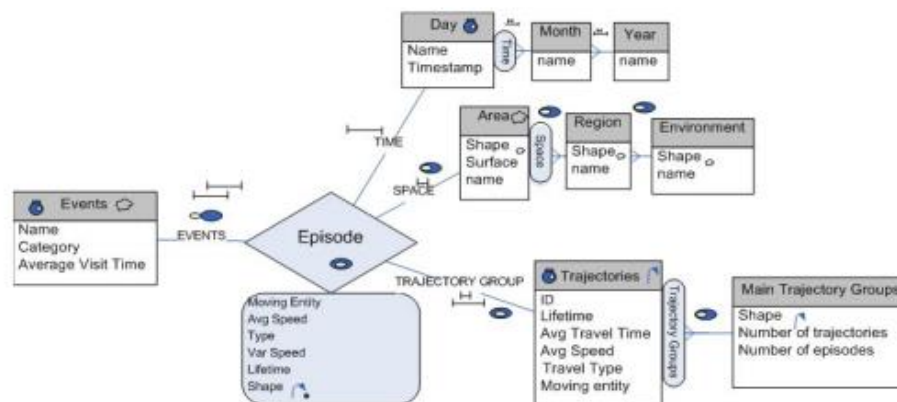
Figura 1-6: Modelo multidimensional para el análisis de mantenimiento de carreteras (Highway maintenance) (Malinowski & Zimanyi, 2005).



De esta forma, los iconos ayudan a representar con más precisión la situación del mundo real, e.g., un segmento de una carretera (Highway segment, línea) pertenece a una sección de una carretera (Highway section, conjunto de líneas) y esta a su vez a una carretera (Highway, conjunto de líneas); una ciudad (superficie) está incluida en un estado (conjunto de superficies). Por su parte, la relación topológica intersección en el hecho (Highway maintenance), establece que entre la ciudad y el segmento de una carretera asociados con un mantenimiento debe haber intersección.

En Campora et al.,(2011) presentan un marco de trabajo (framework) para el diseño de BD que incluyan conceptos espacio-temporales. Se presenta como caso de estudio una BD de trayectorias para el análisis del tráfico, donde se consideran las relaciones espaciales entre las trayectorias y los sitios de interés (e.g., restaurantes, supermercados, entre otros). Para los autores una trayectoria no es solo un conjunto de puntos espacio-temporales, sino un objeto semántico con identidad, el cual contiene sub-objetos semánticos denominados episodios (e.g., detenciones (stops) en los sitios de interés y desplazamientos (moves) entre estos sitios)(Yan et al., 2010). La Figura 1-7 muestra el esquema propuesto por los autores donde se usa la notación de Malinowski y Zimányi (2005; 2005b).

Figura 1-7: BD de episodios (Episodes)(Campora et al., 2011).








Además de las relaciones topológicas presentadas en la Figura 1-5b, se incluyen iconos (en forma de segmentos de recta) para representar relaciones temporales (e.g., el tiempo de ocurrencia de un episodio (Episode) debe coincidir total o parcialmente con el tiempo de ocurrencia de un evento (Event); el tiempo de ocurrencia de un episodio debe estar

dentro del tiempo de ocurrencia de una trayectoria (Trajectory)) e iconos para representar las trayectorias.

En Oueslati y Akaichi (2014) se crea un perfil UML (un perfil UML permite personalizar modelos UML para dominios particulares) llamado Trajectory-UML con el objetivo de mejorar el modelamiento de trayectorias y de BD de trayectorias. Se incluyen estereotipos, iconos y restricciones del diagrama de clases y de secuencia de UML.

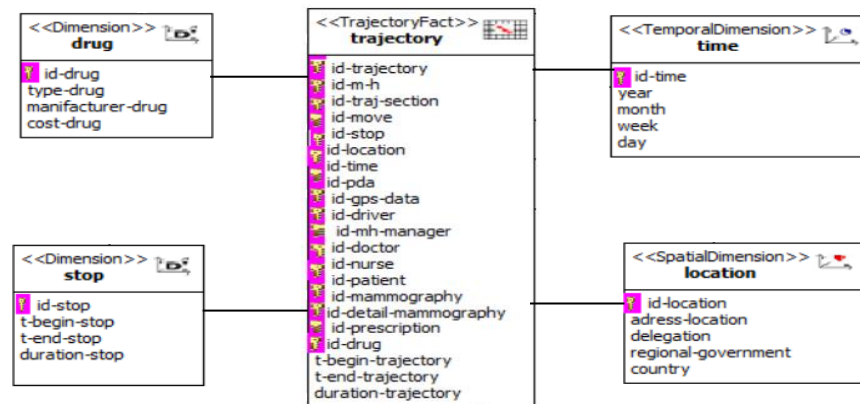
Los estereotipos son elementos de texto que permiten definir las categorías (del dominio) de los objetos de un modelo. La Figura 1-8 muestra los estereotipos e iconos usados. Estas combinaciones facilitan, e.g., la distinción entre los diferentes tipos de dimensiones (espacial, temporal).

Figura 1-8: Estereotipos e iconos usados en el esquema de una BD (Oueslati & Akaichi, 2014).

<i>Stereotypes</i>	<i>Icons</i>
«trajectoryFact»	
<<Dimension>>	
<<TemporalDimension>>	
<<SpatialDimension>>	
<<DimensionLevel>>	

La Figura 1-9 muestra un (fragmento) de un modelo (sobre trayectorias de ambulancias), que combina el diagrama de clases de UML con los estereotipos e iconos mostrados.

Figura 1-9: Fragmento de un modelo conceptual para una BD de trayectorias de ambulancias (Oueslati & Akaichi, 2014).



Jenhani y Akaichi (2014) extienden el trabajo de (Oueslati & Akaichi, 2014) con estereotipos e iconos para representar conceptos como episodios, detenciones (stops), desplazamientos (moves), objetos en movimiento (ambulancias), actores (personal médico) y sitios de interés; véase la Figura 1-10.

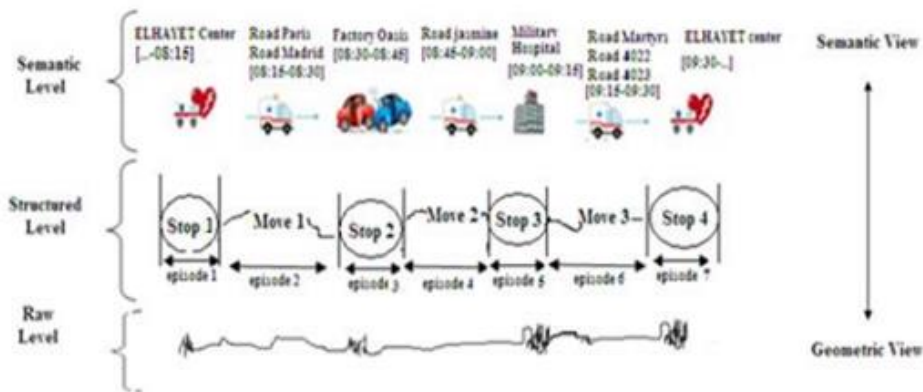
Figura 1-10: Extensiones para la representación semántica de trayectorias (Jenhani & Akaichi, 2014).

Element	Stereotype	Icon
Trajectory	<<Trajectory_Fact>>	
Ambulance	<<Moving_Object_Dim>>	
Episode	<<Episode_Dim>>	
Stop	<<Stop_Dim>>	
Move	<<Move_Dim>>	
Date	<<Time_Dim>>	
Patient, Doctor, Nurse	<<Actor_Dim>>	
Point of interest, Medical institution, Break-down point, Amb_services center, Location of incident	<<PoinOfInterest_Dim>>	
Roads Network	<<LineOfInterest_Dim>>	
Region	<<RegionOfInterest_Dim>>	
Crew, Medical staff,	<<Dimension>>	

Además, se representa una trayectoria mediante tres niveles de abstracción, semántico (semantic), estructurado (structured) y no-refinado (raw). Cada nivel permite capturar

características del mundo real que deben ser consideradas en los modelos conceptuales; un ejemplo se muestra en la Figura 1-11.

Figura 1-11: Representación en niveles de abstracción de una trayectoria (Jenhani & Akaichi, 2014).



El primer nivel es el no-refinado, es el más abstracto. Allí, la trayectoria se percibe como un conjunto de puntos espaciales (x, y) asociados con un tiempo (t).

El segundo nivel es el estructurado, es el puente (nivel intermedio) entre el nivel no-refinado y el nivel semántico. Aquí, la trayectoria se percibe como un conjunto de episodios (detenciones (stops) y desplazamientos (moves)), según la propuesta de (Spaccapietra et al., 2008).

El tercer nivel es el semántico. Aquí, se especifica para cada episodio el espacio (geográfico) donde ocurre; además se especifica el tiempo, los objetos implicados (e.g., ambulancias) y las actividades (e.g., recoger un herido, llegada a un hospital) ejecutadas.

En Arigon et al., (2006) se plantea un modelo multidimensional para datos multimediales (imágenes o señales) multiversionando las dimensiones (i.e., para una misma dimensión existen varias versiones de esta; esto permite comparar las medidas) En los modelos conceptuales para BD, los autores usan la representación multimedial del objeto a analizar en los hechos como se observa en la Figura 1-12, donde se usa un icono en el interior del rombo para representar un electrocardiograma. Otro ejemplo se muestra en la Figura 1-13, donde se usa un icono representativo del genoma humano, además se incluyen

polígonos para abarcar los niveles de cada una de las dimensiones, lo cual facilita su identificación.

Figura 1-12: Modelo conceptual para una BD de electrocardiogramas (Arigon et al., 2006).

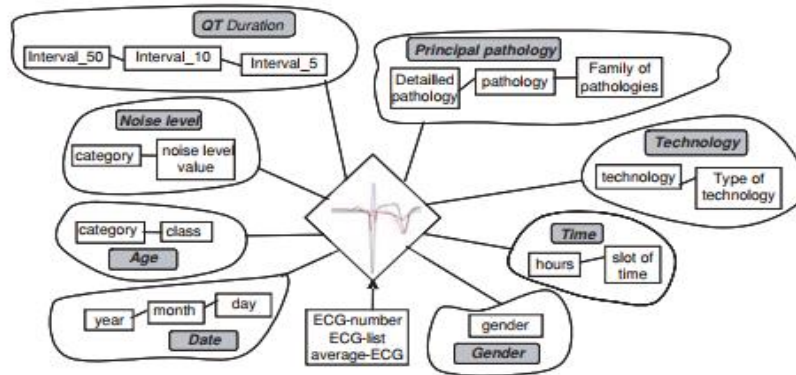
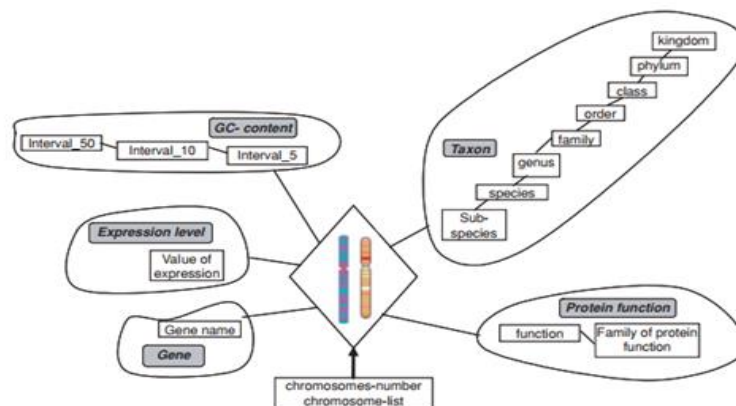


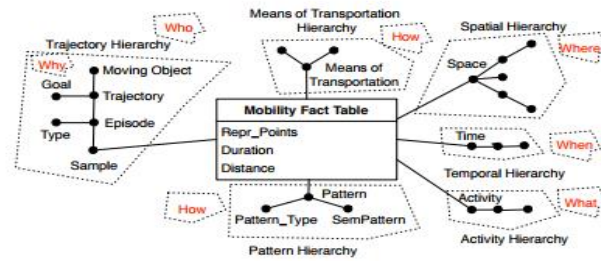
Figura 1-13: Modelo conceptual para una BD de bioinformática (Arigon et al., 2006).



En Wagner et al.,(2014) se propone un método narrativo llamado 5W1H el cual explica sistemáticamente el contexto que involucra un problema. El método plantea seis preguntas narrativas (narrative questions, Who, What, When, Where, Why y How) por medio de las cuales se trata de modelar un problema. El método está orientado al modelamiento de trayectorias. Cada pregunta narrativa del modelo 5W1H se correlaciona con una característica específica de una trayectoria. Así, un objeto (Who) se puede desplazar mediante algún medio de transporte o tener algún comportamiento (How) y este ejecuta una actividad (What), por una razón (Why), en un momento (When) y en un lugar (Where). Los autores afirman que su método facilita la formulación de consultas sobre el movimiento de los objetos (e.g., ¿cuál es la distancia de viaje promedio de las personas que usan el

transporte público y visitan al menos una atracción cultural?, ¿cuáles son las trayectorias de automóviles que poseen una velocidad media menor a 30 km/h y que se encuentran en una congestión?).

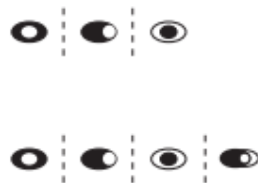
Figura 1-14: Modelo conceptual para una BD de trayectorias con 5W1H (Wagner et al., 2014).



La Figura 1-14 presenta un modelo conceptual enriquecido para una BD. En este, las dimensiones se asocian con las preguntas narrativas que brindan la semántica de la trayectoria (e.g., el tiempo se representa con la pregunta When y el espacio con la pregunta Where) y son delimitadas con polígonos punteados para facilitar su identificación; además cada nivel se representa con un punto.

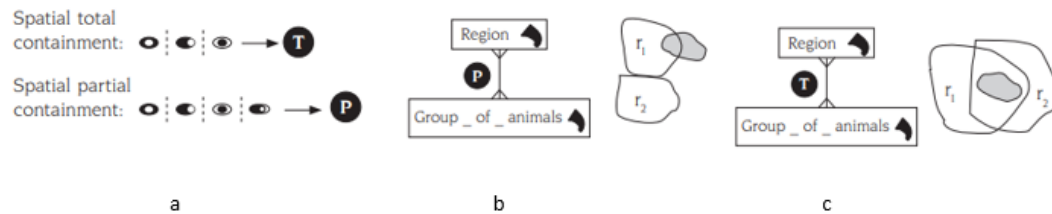
En Moreno et al.,(2010) se enriquece un modelo conceptual multidimensional espacial, con elementos temporales, convirtiéndolo en un modelo conceptual espacio-temporal. Los autores extienden la notación de (Malinowski & Zimanyi, 2005) mediante las restricciones topológicas de la Figura 1-15.

Figura 1-15: Disyunción exclusiva en relaciones espaciales En (Moreno et al., 2010)



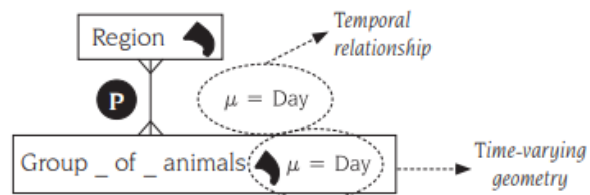
Las líneas verticales punteadas representan una disyunción exclusiva de las relaciones espaciales. Adicionalmente, se proponen símbolos para representar el tipo de inclusión entre los niveles espaciales, i.e., *inclusión total o parcial* (Jensen et al., 2002; Moreno et al., 2010) como se muestra en la Figura 1-16.

Figura 1-16: Tipos de inclusión y disyunción, a) inclusión total y parcial, b) ejemplo de inclusión parcial y c) ejemplo de inclusión total En (Moreno et al., 2010).



También se incorporan iconos para representar aspectos temporales como se muestra en la Figura 1-17.

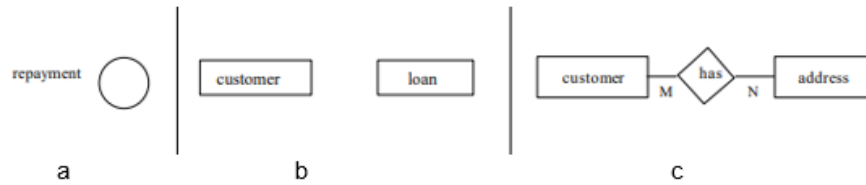
Figura 1-17: Relación temporal y geometría variable en el tiempo En (Moreno et al., 2010).



Así, e.g., el polígono en Group_of_animals indica que es un nivel espacial (el polígono representa el área cubierta por un grupo de animales) y $\mu = \text{Day}$ (ubicado al lado del polígono) indica que el área cubierta varía y que dicha variación se puede registrar diariamente. El círculo con la P entre los niveles indica que existe una inclusión parcial entre el área cubierta por un grupo de animales y la región (Region), i.e., el polígono del grupo de animales puede intersectar el polígono de la región. $\mu = \text{Day}$ (ubicado en la relación) indica que el grado (porcentaje) de inclusión varía y que dicha variación se puede registrar diariamente.

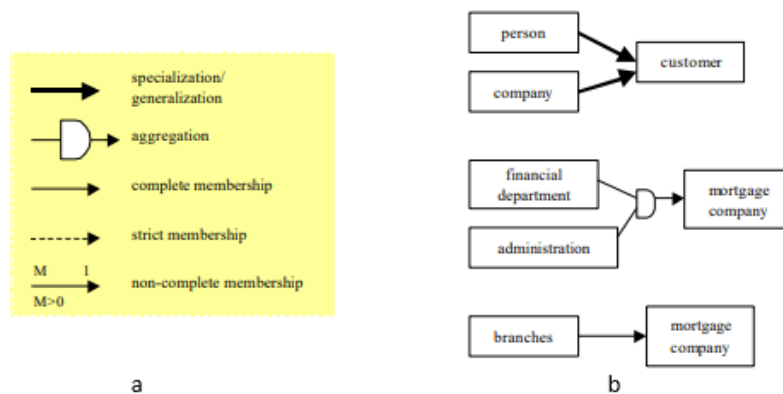
En Tryfona et al., (2004) se combina el esquema en estrella con elementos del modelo ER (Entidad–Relación) (Chen, 1976). Los autores adaptan los elementos del ER para el diseño de una BD. Los hechos se representan con círculos, los niveles con rectángulos, las relaciones entre niveles y hechos o entre niveles con rombos y poseen cardinalidad así, muchos a muchos (N:M), muchos a uno (M:1) o uno a muchos (1:M), véase la Figura 1-18.

Figura 1-18: Representación de elementos de una BD con elementos del modelo ER: a) hecho, b) niveles-entidades y c) Relaciones entre entidades (Tryfona et al., 2004).

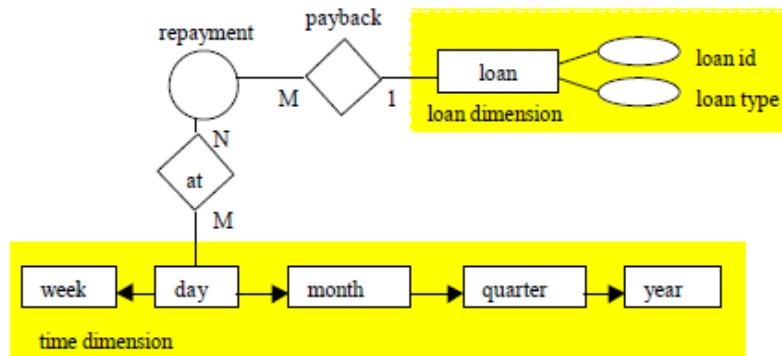


Las relaciones entre las entidades pueden ser de tipo especialización/generalización, agregación (i.e., objetos como partes de un objeto, e.g., una compañía se divide en departamentos) y miembro (i.e., un objeto es miembro de otro y tiene las mismas características y comportamiento de este, e.g., las sucursales (branches) de una compañía son miembros de la compañía). En la Figura 1-19 se presentan ejemplos de estos tipos de relaciones y su representación gráfica.

Figura 1-19: Relaciones entre entidades, (a) representación y (b) ejemplos (Tryfona et al., 2004).



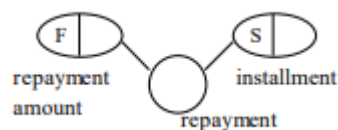
La Figura 1- 20 muestra un fragmento del modelo propuesto por los autores, construido a partir de una empresa de hipotecas (mortgages).

Figura 1- 20: Hechos, dimensiones y jerarquías StarER (Tryfona et al., 2004).

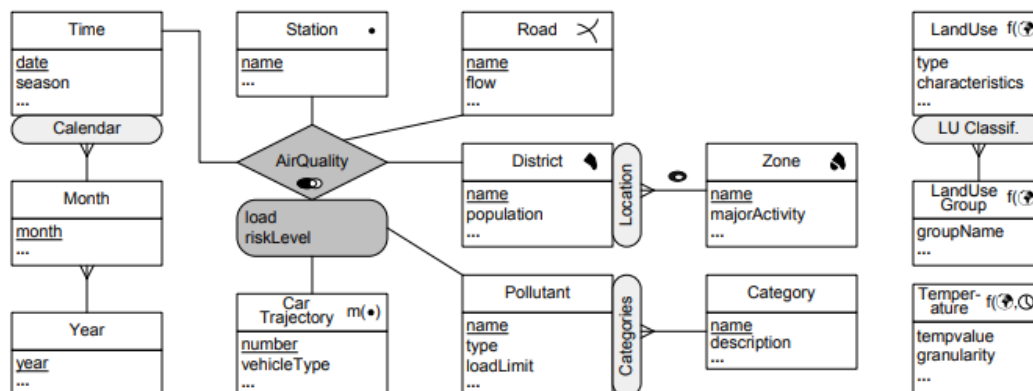
Como se observa en la Figura 1- 20 las entidades y los hechos poseen atributos, estos se representan con óvalos. Los atributos de los hechos suelen ser valores numéricos y se clasifican en tres tipos:

- Medida (S): valores cuya agregación (suma, promedio) tiene sentido para el negocio (e.g., cantidad vendida).
- Flujo (F): valores cuya agregación (suma, promedio) tiene sentido para el negocio y se aplican sobre periodos registrando el cambio en el tiempo (e.g., monto de reembolso).
- Valor por unidad (V): valores cuya agregación (suma, promedio) no tiene sentido para el negocio (e.g., tasas de interés).

Un ejemplo se muestra en la Figura 1-21.

Figura 1-21: Tipos de atributos de hechos y representación gráfica (Tryfona et al., 2004).

En Zimányi (2012) se presenta un caso de estudio para el análisis de variables meteorológicas (e.g., temperatura, calidad del aire) mediante una BD espacio-temporal. Se extiende la notación presentada en (Malinowski & Zimányi, 2008). La Figura 1-22 muestra el modelo propuesto por los autores el cual consta de un hecho llamada AirQuality y ocho dimensiones (dos de ellas no relacionadas con el hecho).

Figura 1-22: Modelo Espacio – temporal (Zimányi, 2012).

En este modelo se considera el concepto de moving region (región en movimiento), i.e., se considera el movimiento de objetos en crecimiento (e.g., huracanes incendios forestales; confrontar con la propuesta de Moreno et al.,(2010)). También se consideran funciones de tipos de movimientos denotadas $m(.)$ las cuales representan puntos o áreas en movimiento (i.e., objetos móviles). Además, se incluyen funciones para representar fenómenos que varían con el tiempo o en el espacio (e.g., la temperatura) y se representan mediante $f(.)$, donde en los paréntesis se puede incluir un globo terráqueo (variación espacial) o un reloj (variación temporal).

1.3 Comparación y clasificación de modelos para BD

En la Tabla 1-1 se comparan los trabajos presentados en este documento.

Tabla 1-1: Comparación y clasificación de modelos para BD.

Referencia	Dominio	Iconos en los hechos	Iconos en las dimensiones	Iconos en las relaciones	Observaciones
(Malinowski & Zimanyi, 2005)	Geo informática, manejo de trayectorias	✓	✓	✓	Manejo de conceptos geométricos y de relaciones topológicas

Referencia	Dominio	Iconos en los hechos	Iconos en las dimensiones	Iconos en las relaciones	Observaciones
(Campora et al., 2011)	Espacio-temporal, análisis de tráfico	√	√	√	Representación gráfica y gestión de objetos espacio temporales
(Oueslati & Akaichi, 2014)	Logística médica, análisis de trayectorias	√	√		Clasificación de tipos de dimensiones mediante estereotipos
(Jenhani & Akaichi, 2014)	Logística médica, análisis de trayectorias	√	√		Clasificación de tipos de dimensiones y de objetos espacio temporales, semántica en relaciones topológicas
(Arigon et al., 2006)	Bioinformática	√		√	Versionamiento de dimensiones
(Wagner et al., 2014)	Análisis de trayectorias		√	√	Uso de métodos narrativos para el manejo y análisis de trayectorias
En (Moreno, Alberto, Arias, & Losada, 2010)	Espacio-temporal, análisis de trayectorias	√	√	√	Enriquecimiento de relaciones topológicas
(Tryfona et al., 2004)	Sector financiero e inmobiliario	√	√		Adecuación del modelo Entidad-Relación para BD
(Zimányi, 2012)	Análisis de trayectorias, meteorología	√	√	√	Planteamiento de funciones para el análisis de trayectorias de objetos o de áreas de gran tamaño

Las agregaciones pictóricas y conceptuales realizadas sobre los elementos que componen los modelos de BD como dimensiones, hechos y relaciones permiten describir con mayor precisión elementos propios de las distintas áreas de estudio, los diferentes autores incorporan iconos principalmente sobre la tabla de hechos y dimensiones con el objetivo de incrementar la expresividad dentro de estos elementos.

1.4 Área de estudio por explorar

En este capítulo se presentaron y compararon distintos modelos conceptuales para BD que han sido enriquecidos con iconos. La revisión sugiere que no existen, e.g., modelos conceptuales para BD que consideren el manejo de matrices en las dimensiones o en los hechos. Para ello es necesario definir iconos que permitan representar este tipo de elementos y las operaciones (suma, multiplicación de matrices) y restricciones válidas entre ellas (e.g, matrices invertibles). Esto puede ser útil para dominios como la meteorología (Jones & Thornton, 1997), o contabilidad (Goicoechea et al., 1991) la cual transforma el método de contrapartida (Rajadell, Trullàs, & Simo, 2014) ubicando las cuentas del estado financiero en entradas matriciales que facilitan la consolidación final de cada cuenta dentro de periodos establecidos (e.g., Mensual), en meteorología específicamente para representar los datos generados por radares meteorológicos para la descripción de las condiciones meteorológicas de una región, estos instrumentos crean una matriz sobre una zona, calculando para cada punto (latitud, longitud) valores de concentración de hidrometeoros (Ciach & Krajewski, 2002). Al darles a estas mediciones un tratamiento matricial, se podrían obtener indicadores más fáciles de interpretar por parte del usuario final (e.g., promedio de los datos de una matriz).

Aunque en los últimos años se han enriquecido diversos modelos conceptuales para BD hay dominios como el químico, el petrolero, el financiero, el textil, entre otros, que carecen de modelos enriquecidos. Cada uno de estos dominios conlleva particularidades que deben ser consideradas para lograr modelos más precisos.

2. Matrices aplicadas

Diversos campos han encontrado en el álgebra matricial un impulso al desarrollo de su disciplina, permitiendo facilidad y precisión en el manejo de la información, al ser un área del conocimiento bien conocida es posible amalgamar sus propiedades con las particularidades de cada dominio, esto permite crear modelos matemáticos robustos y fácilmente tratados mediante algoritmos o técnicas informáticas.

2.1 Matriz de Leslie

En biología las matrices se suelen usar para proyectar poblaciones, como se muestra en (Bock et al., 2016) donde se construye una “tabla de vida” para las iguanas verdes de Barro Colorado, Panamá. La tabla incluye datos sobre las tasas de fertilidad y de natalidad y otros datos demográficos. En la Tabla 2-1 se muestra un ejemplo.

Tabla 2-1: Parámetros demográficos de iguanas verdes hembras del nido Slothia – Panamá (Bock et al., 2016).

Rango de edad	Número de individuos (X_0)	Tasa de fecundidad	Tasa de supervivencia (supervivencia hasta que se alcanza la progresión al siguiente rango de edad)
0 a 1	1	0	0.039
1 a 2	38,924	0	0.387
2 a 3	15,064	0.008	0.973
3 a 4	14,663	0.154	0.757
4 a 5	11,109	1.062	0.725
5 a 6	8,055	3.116	0.688
6 a 7	5,546	7.825	0.634
7 a 8	3,521	12.16	0.593
8 a 9	2,089	13.11	0

Además, se crea una matriz de proyección poblacional usada en el análisis de sensibilidad y elasticidad poblacional lo que permite determinar las tasas vitales que afectan la tasa de crecimiento poblacional.

Para ello, en Bock et al., (2016) se usa una matriz conocida como matriz de Leslie que evalúa el crecimiento de una población mediante su tasa de supervivencia agrupada por rangos de edad (Hearon, 1976). La matriz de Leslie está compuesta así:

La primera fila se compone de las tasas de fecundidad de la especie por rangos de edad (cada columna representa un rango de edad). Las siguientes filas están compuestas por ceros excepto la diagonal menor inferior que contiene las tasas de supervivencia de cada rango de edad como se muestra en la Figura 2-1.

Figura 2-1: Matriz de Leslie (Bock et al., 2016).

$$A = \begin{pmatrix} 0.000 & 0.000 & 0.008 & 0.154 & 1.062 & 3.116 & 7.825 & 12.16 & 13.11 \\ 0.039 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.387 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.973 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.757 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.725 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.688 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.634 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.593 & 0.000 \end{pmatrix}$$

Las condiciones iniciales como tasas o número de población X_0 se obtienen a partir de estudios hechos en la zona y técnicas de análisis de individuos como (captura, marcado y recaptura). Cuando un individuo es capturado se obtienen datos como su edad, tamaño y es marcado con el fin de identificarlo en el momento de una eventual recaptura.

Para conocer la proyección poblacional del siguiente periodo $X_0 + 1$ (i.e., los nuevos individuos pertenecientes a los diferentes rangos) se multiplica la matriz de Leslie inicial A con el vector compuesto por el número de individuos por rangos de edad mostrado en la Tabla 2-1, como se muestra en la ecuación $X_1 = A * X_0$ (2. 1).

$$X_1 = A * X_0. \quad (2. 1).$$

Los datos de una matriz están asociados con variables como el territorio, la temporada del año, el sitio de anidamiento, el rango de edad de las especies. Esto evidencia dimensiones comúnmente usadas en una BD.

2.2 Matrices en imágenes digitales

En Prabhune et al., (2017) se usan matrices para el procesamiento digital de imágenes, lo que permite la extracción, el mejoramiento y el filtrado de datos de la imagen, mediante operaciones matriciales ya que una imagen digital se puede representar como una matriz que almacena los datos correspondientes a cada pixel (color del pixel e intensidad del color).

Considerando el método de descomposición matricial (Kalman, 2002), el cual permite expresar cualquier matriz en términos de otras matrices especiales como, matriz ortogonal, unitaria, diagonal, triangular entre otras, una matriz A se puede expresar como la multiplicación de tres matrices.

En una imagen se puede incluir, e.g., una marca de agua (símbolo o texto como el nombre o el logo de una compañía). Esta marca ayuda a comprobar la autenticidad de la imagen. La descomposición en matrices es un método comúnmente usado para obtener o incluir marcas de agua en una imagen.

Los métodos más comúnmente usados para incluir marcas de agua en una imagen requieren diversas operaciones matriciales:

modificaciones a las matrices de la descomposición, multiplicaciones de matrices, trasposiciones, entre otras. Este ejemplo, sugiere de nuevo dimensiones como tipo de imagen o autor del contenido multimedia.

2.3 Contabilidad Matricial

Un hecho contable representa un movimiento financiero en una organización, e.g., los movimientos monetarios hechos para la compra de materia prima o para la venta de productos. Los hechos contables se registran en cuentas (contables) tales como Inventario, Banco, Cuentas x Pagar, entre otras. Gracias a la metodología de contrapartida, creada por Luca Pacioli (Rajadell et al., 2014), para la gestión de estas cuentas, es posible llevar el control de los estados financieros de una organización.

2.3.1 Contabilidad clásica (contrapartida)

En esta metodología se inscriben todas las cuentas en el libro mayor (Rajadell et al., 2014) mediante una contrapartida, que visualmente semeja a la letra T, donde en la columna izquierda se anotan los débitos y en la columna derecha los créditos, ver Figura 2-2 (los débitos y los créditos son componentes de las cuentas T, los cuales representan inscripciones monetarias que corresponden a un aumento o a una disminución del saldo de una cuenta según su tipo).

Figura 2-2: Ejemplo de cuenta T para la cuenta de activo Banco.

Banco Cuenta de Activo	
Débito	Crédito
	5000

Para determinar donde se debe anotar un hecho contable (en el crédito o en el débito), se debe identificar la cuenta correspondiente y su tipo, i.e., si esta representa una cuenta de activo, de pasivo, de ingresos, de costo, de patrimonio o de gastos como se muestra en la Tabla 2-2.

Tabla 2-2: Tipos de cuentas (Lucía Urueña & San Mateo, 2010).

Tipo de cuenta	Aumenta	Disminuye
Activo	Débito	Crédito
Pasivo	Crédito	Débito
Patrimonio	Crédito	Débito
Ingresos	Crédito	Débito
Costos	Débito	Crédito
Gastos	Débito	Crédito

Por ejemplo, un aumento en una cuenta de activo se apunta en la columna débito y un aumento en una cuenta de pasivo se apunta en la columna crédito.

De esta forma todo movimiento (aumento o disminución) en una cuenta se ve reflejado como una contrapartida o movimiento inverso en otra(s) cuenta(s).

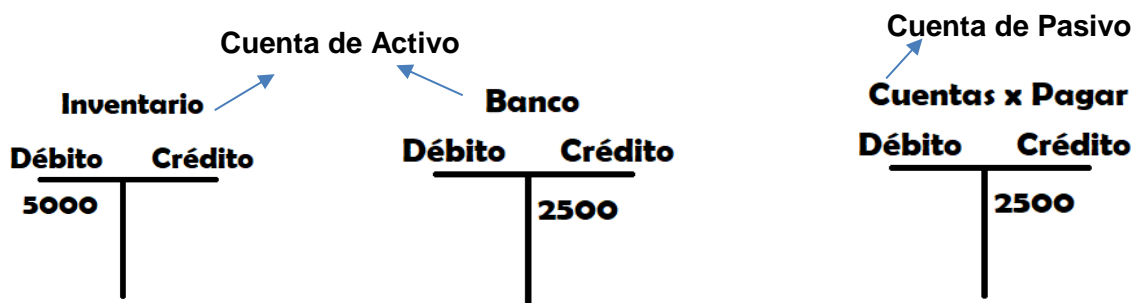
2.3.2 Ejemplo Contrapartida

Inicialmente, se deben listar las cuentas principales, conocidas como cuentas del estado financiero, las cuales consolidan información de periodos establecidos: Cuentas que incrementan a) con el débito: Activo y Egresos y b) con el crédito: Pasivo, Ingresos y Patrimonio.

Para la consolidación de estas cuentas se requieren algunas cuentas intermedias, las cuales llevan el detalle de los movimientos como se muestra a continuación.

supóngase que el día 1 se compra materia prima por valor de \$5000; se paga con cheque el 50% y el resto se pagará a 30 días. Esta compra aumentará la cuenta inventario (cuenta de Activo) en \$5000, disminuirá la cuenta Banco (cuenta de Activo) como contrapartida en \$2500 pesos y aumentará la cuenta Cuentas x Pagar (cuenta de Pasivo) en \$2500, como se observa en las respectivas cuentas T, véase la **Figura 2-3**.

Figura 2-3: Cuentas T- Estado inicial.



Como se observa en la **Figura 2-3**, un hecho contable involucra *al menos dos cuentas* para reflejar la contrapartida, i.e., conlleva un movimiento monetario entre dos o más cuentas. El día 30 al pagar el 50% restante, la cuenta Banco disminuye en \$2500 correspondientes al 50% faltante y la cuenta Cuentas x Pagar aumenta en \$2500 consolidándose así en \$0, i.e., no se tienen deudas y la cuenta Banco se consolida con una disminución total de \$5000 como se observa en la Figura 2-4.

Figura 2-4: Cuentas T - Estado final.

Banco		
Débito	Crédito	
	2500	
	2500	
0	5000	= -5000

Cuentas x Pagar		
Débito	Crédito	
2500	2500	
2500	2500	= 0

2.3.3 Contabilidad Matricial

En contabilidad se considera la contrapartida como una metodología clásica para llevar el control de los estados financieros. Esta metodología ha sido modificada por autores como (Goicoechea López-Vailo et al., 1991), donde se combina la metodología clásica y el álgebra matricial, y se reemplazan las cuentas T por una matriz donde se inscriben todas las cuentas, véase la Figura 2-5.

Figura 2-5: Matriz de cuentas (Poca Freddy, 2017).

	Caja	Capital	Gastos Generales	Sueldos y salarios	Inventario de Mercaderías	Banco Nacional	Muebles y Enseres	Compras	Cuentas por Pagar	Cuentas por Cobrar	Ventas	Equipo de Computación	Vehículo	Energía Eléctrica y Agua	Servicio Telefónico	Documentos por Pagar	Perdidas y Ganancias
Caja																	
Capital																	
Gastos Generales																	
Sueldos y salarios																	
Inventario de Mercaderías																	
Banco Nacional																	
Muebles y Enseres																	
Compras																	
Cuentas por Pagar																	
Cuentas por Cobrar																	
Ventas																	
Equipo de Computación																	
Vehículo																	
Energía Eléctrica y Agua																	
Servicio Telefónico																	
Documentos por Pagar																	
Perdidas y Ganancias																	

Así, se facilita la relación entre las diferentes cuentas al inscribir un hecho contable. Para esto se deben considerar algunas convenciones: en (Goicoechea López-Vailo et al., 1991;

Poca Freddy, 2017) se propone llamar débitos a las filas y créditos a las columnas, la intersección fila – columna (débito-crédito) representa el movimiento de un hecho contable. Además, a diferencia de la metodología de contrapartida, aquí un movimiento entre dos cuentas se inscribe una sola vez.

Así, los movimientos del ejemplo anterior se pueden representar mediante la matriz de la Tabla 2-3.

Tabla 2-3: Ejemplo de matriz de cuentas.

	Caja	Banco	Inventario	Cuentas X Pagar
Caja				
Banco			2500	2500
Inventario				
Cuentas X Pagar			2500	

Al totalizar las filas y las columnas de la Tabla 2-3, se obtienen dos vectores de consolidación de créditos y de débitos, véase la Tabla 2-4.

Tabla 2-4: Consolidación: vectores de créditos y de débitos.

	Caja	Banco	Inventario	Cuentas X Pagar	Vector de Créditos
Caja	0	0	0	0	0
Banco	0	0	2500	2500	5000
Inventario	0	0	0	0	0
Cuentas X Pagar	0	0	2500	0	2500
Vector de Débitos	0	0	5000	2500	

Mediante operaciones del álgebra matricial es posible consolidar las cuentas: primero se transpone uno de los dos vectores, así ambos serán vectores columna o vectores fila; posteriormente, se calcula el saldo mediante una resta de vectores (Saldo final = Débitos – Créditos) y se obtienen los saldos finales para cada cuenta como se muestra en la Tabla 2-5.

Tabla 2-5: Vectores de débitos, de créditos y de saldos finales.

	Vector de Débitos	Vector de Créditos	Vector de Saldos Finales
Caja	0	0	0
Banco	0	5000	-5000
Inventario	5000	0	5000
Cuentas X Pagar	2500	2500	0

El álgebra matricial ahorra esfuerzo al inscribir y consolidar las cuentas; además, su gestión se facilita mediante un lenguaje de programación o de programas contables.

2.4 Bases de datos matriciales

En Luo et al., (2018) son incorporados tipos de datos y operadores para soportar operaciones de álgebra lineal en una base de datos relacional. Los tipos de datos incluyen escalar (LBELED_SCALAR), vector (VECTOR) y matriz (MATRIX). Los operadores incluyen entre otros suma y multiplicación de matrices, transposición, multiplicación de una matriz por un escalar, entre otros.

Esta incorporación implica pocos cambios en un sistema de gestión de bases de datos relacional (SGBDR). Aunque los SGBDRs suelen incluir un tipo de datos arreglo (el cual se puede usar para representar un vector), e.g., el tipo VARRAY de Oracle, usualmente carecen de operadores (nativos) de álgebra lineal para su gestión.

La formulación en SQL de determinadas operaciones del álgebra lineal puede llevar a soluciones complejas y de difícil optimización por parte del optimizador. Una opinión similar se expresa en Celko (SQL for Smarties).

supóngase que se tiene el vector $x = \{x_1, x_2, \dots, x_n\}$ y una matriz A y se desea calcular la distancia entre estos como denota la ecuación $d_A^2(x_i, x') = (x_i - x')^T A (x_i - x')$

(2. 2).

$$d_A^2(x_i, x') = (x_i - x')^T A (x_i - x') \quad (2. 2).$$

La ecuación $d_A^2(x_i, x') = (x_i - x')^T A (x_i - x')$

(2. 2). puede expresarse SQL como se muestra en la Figura 2-6.

Figura 2-6: Consulta distancia.

```
CREATE VIEW xDiff (pointID, dimID, value) AS
SELECT x2.pointID, x2.dimID, x1.value - x2.value
FROM data AS x1, data AS x2
WHERE x1.pointID = i and x1.dimID = x2.dimID

SELECT x.pointID, SUM (firstPart.value * x.value)
FROM (SELECT x.pointID AS pointID, a.colID AS
      colID, SUM (a.value * x.value) AS value
      FROM xDiff AS x, matrixA AS a
      WHERE x.dimID = a.rowID
      GROUP BY x.pointID, a.colID)
      AS firstPart, xDiff AS x
WHERE firstPart.colID = x.dimID
      AND firstPart.pointID = x.pointID
GROUP BY x.pointID
```

Donde la tabla data tiene los atributos: pointID, dimID, value; y la tabla matrixA: rowID, colID y value.

Aunque en SQL es factible formular esta consulta de otras formas, estas son posiblemente similares en complejidad.

Con la propuesta de los autores la formulación de la consulta se simplifica, es más natural y facilita además el trabajo del optimizador de consultas como se muestra en la Figura 2-7.

Figura 2-7: Consulta distancia optimizada.

```
SELECT x2.pointID,
      inner_product (
        matrix_vector_multiply (
          a.val, x1.val - x2.val),
          x1.val - x2.val) AS value
FROM data AS x1, data AS x2, matrixA AS a
WHERE x1.pointID = i
```

En total se incorporaron 22 operadores para la gestión de los nuevos tipos de datos. Entre estas están la extracción de la diagonal de una matriz (`diag()`), la multiplicación de una matriz por un vector (`matrix_vector_multiply()`) y la multiplicación de matrices (`matrix_multiply()`).

A continuación, se muestra un ejemplo. Sea la tabla `m` con dos atributos, uno de tipo `MATRIX` y otro de tipo `VECTOR`.

- `CREATE TABLE m (mat MATRIX [10] [10], vec VECTOR [100]);`

Sea la consulta:

- `SELECT matrix_vector_multiply (m.mat, m.vec) AS res FROM m;`

En este punto el SGBD debe validar la compatibilidad de las operaciones (e.g., que el número de columnas de la matriz coincida con el número de elementos del vector) e indicar el error como ocurriría en este ejemplo.

La propuesta se implementó en SimSQL (Cai et al., 2013) (SimSQL es un SGBDR que soporta paralelismo y ejecuta sobre Hadoop) y se comparó con otras plataformas como SystemML V0.9 (The Apache Software Foundation, 2015), SciDB V14.8 (Paradigm4, 2019) y Spark V1.6 mllib.linalg (i.e., la librería de álgebra lineal de Spark),

SimSQL es evaluado en 3 tipos de implementaciones, Tuple SimSQL (i.e. SimSQL original sin las mejoras e implementaciones de los objetos Vector/matriz propuestos por los autores), Vector SimSQL (e.i. almacenamiento de los datos en vectores) y Block SimSQL (i.e. datos almacenados como vectores y luego agrupados en bloques, incluyendo el atributo `block_index`, el cual identifica cada bloque).

Los experimentos incluyeron:

- Cálculo de la matriz de Matriz de Gram $G = XTX$ ver Figura 2-8.
- Regresión lineal por mínimos cuadrados ver Figura 2-9.

- Detección de valores atípicos (outliers) en un espacio vectorial ver Figura 2-10.

Figura 2-8: Resultados cálculo de la matriz de Gram (Luo et al., 2018).

Gram Matrix Computation			
Platform	10 dims	100 dims	1000 dims
Tuple SimSQL	00:01:28	00:03:19	05:04:45
Vector SimSQL	00:00:37	00:00:43	00:05:43
Block SimSQL	00:01:18	00:01:23	00:02:53
SystemML	00:00:05*	00:00:51	00:02:34
Spark mllib	00:00:20	00:00:54	00:17:31
SciDB	00:00:03	00:00:17	00:03:20

Figura 2-9: Resultados regresión Lineal (Luo et al., 2018)..

Linear Regression			
Platform	10 dims	100 dims	1000 dims
Tuple SimSQL	00:03:42	00:05:46	05:05:22
Vector SimSQL	00:00:45	00:00:49	00:06:35
Block SimSQL	00:02:23	00:02:22	00:04:22
SystemML	00:00:06*	00:00:53	00:02:38
Spark mllib	00:00:35	00:01:01	00:17:42
SciDB	00:00:15	00:00:33	00:06:04

Figura 2-10: Resultados detección de Outlier (Luo et al., 2018)..

Outlier Detection			
Platform	10 dims	100 dims	1000 dims
Tuple SimSQL	Fail	Fail	Fail
Vector SimSQL	00:10:14	00:11:49	00:13:53
Block SimSQL	00:03:14	00:04:43	00:10:36
SystemML	00:13:29	00:22:38	00:33:22
Spark mllib	01:22:59	01:15:06	01:13:06
SciDB	00:03:46	00:04:54	00:05:06

De la Figura 2-8, Figura 2-9, Figura 2-10 se observa un buen rendimiento de la implementación Vector SimSQL en contraste con herramientas especializadas para el manejo de objetos matriciales como SystemML y SciDB.

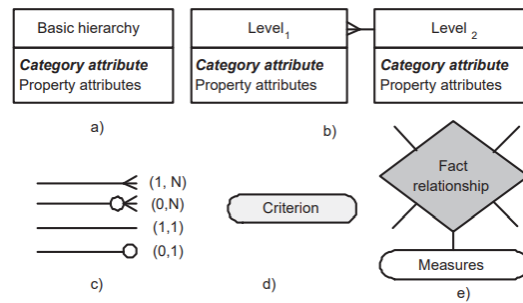
3. Bodegas de datos con objetos matriciales

El álgebra matricial se puede usar junto con las bodegas de datos (BDs) y de esta forma aprovechar las operaciones disponibles de estas dos áreas. Por ejemplo, esta combinación puede ayudar a consolidar las cuentas (como se explicó en el capítulo 2.3) mediante la creación de vectores o de matrices que agreguen los hechos contables, en esta capítulo son propuestos elementos pictóricos que enriquecen modelos de BD convencionales y así describir con mayor precisión las necesidades particulares de cada área de estudio como la contabilidad matricial, inicialmente se realiza un contraste de un modelo de BD para el manejo de hechos contables de forma convencional y un modelo de BD para el manejo de hechos contables con la notación matricial propuesta, esta notación abarca objetos pertenecientes al álgebra matricial, operaciones y restricciones propias de estas áreas.

3.1 Bodega de datos convencional hechos contables

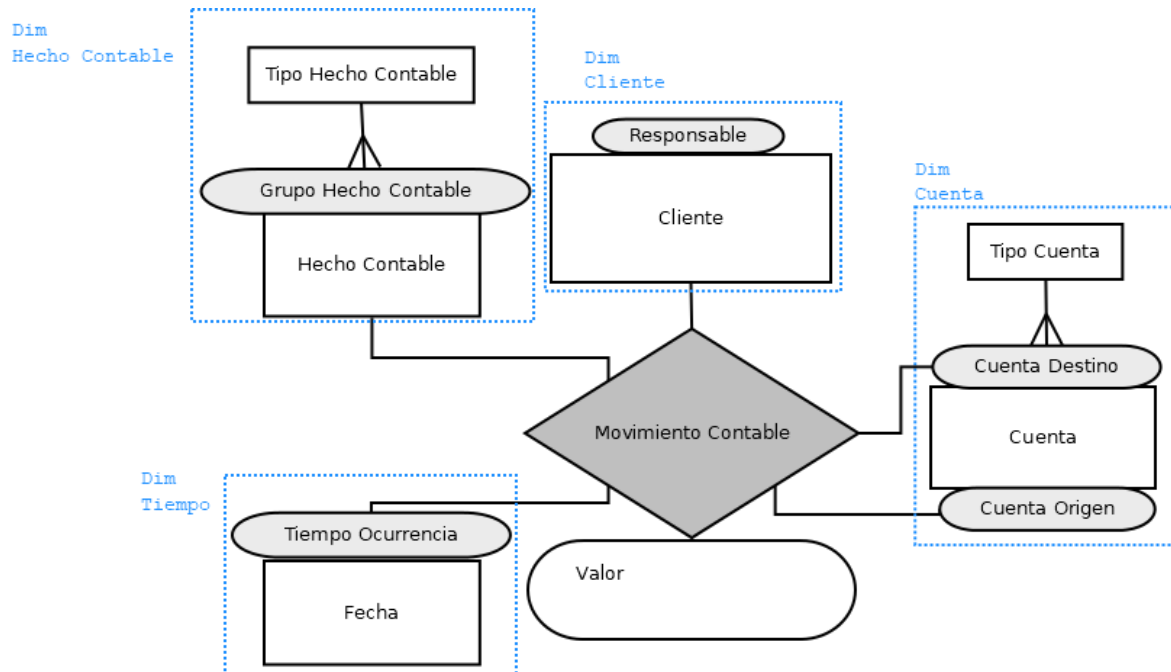
Utilizando los elementos esenciales de la notación de Malinowski y Zimanyi (Malinowski & Zimányi, 2008) para el modelamiento conceptual de BDs (ver Figura 3- 1) convencionales es planteado el modelo mostrado en la Figura 3- 2, el cual modela movimientos contables como los realizados por una empresa en la cotidianidad de sus transacciones como compra y venta de materiales.

Figura 3- 1: Notación de Malinowski y de Zimanyi (Malinowski & Zimányi, 2008): a) dimensión con jerarquía básica (un solo nivel), b) dimensión con jerarquía compuesta por dos niveles, c) cardinalidades, d) criterio de análisis y e) hechos y medidas (measures).



En la Figura 3- 2 se presenta un modelo conceptual para una BD para hechos contables donde se usa dicha notación.

Figura 3- 2: Modelo conceptual para una BD de movimientos de hechos contables.



En esta BD se pueden plantear consultas como:

- A). ¿Cuál es el saldo de las cuentas que tuvieron movimientos el 4 y 6 de mayo de 2018?
- B). ¿Cuál es el saldo de la cuenta x pagar asociada con el cliente X el mes de Mayo?
- C). ¿Cuál es el saldo o estado final de todas las cuentas el día 30 del mes anterior?

- D). Si se consolidan los créditos de todas las cuentas del mes actual y las del mes anterior, al compararlas, ¿qué resultados se obtienen?

Nótese que para resolver la consulta d), es necesario i) obtener los créditos de todas las cuentas del mes actual, i.e., obtener de las cuentas origen (ver modelo de la Figura 3- 2) el valor y sumar estos valores agrupados por cuenta, ii) hacer el mismo cálculo para las cuentas del mes anterior y iii) comparar los resultados de i) y ii).

Por otro lado, si los datos se modelasen mediante tipos de datos como vectores y matrices, se podría simplificar la solución de algunas consultas, como la anterior.

Para la elaboración del nuevo modelo se deben identificar las medidas que se desean gestionar en la BD de forma matricial.

3.2 Ejemplo movimientos de hechos contables

Considérense los siguientes movimientos financieros:

- El 1 de mayo se inicia la actividad económica de la empresa Nutrition S. El accionista principal aporta en efectivo \$200000 y un aporte en mercancía para almacén por \$380000.
- El 4 de mayo se vende mercancía por \$289000 al cliente Project Fitness, el cual paga en efectivo \$79000 y mediante transferencia bancaria \$100000; además, entrega como forma de pago una máquina elíptica evaluada en \$60000 y firma un pagaré por el resto (\$50000) a 10 días.
- El 5 de mayo se compran muebles para oficina al proveedor Galería V por \$5000, los cuales se pagan así: \$2500 mediante transferencia bancaria y el 50% restante serán pagados a 20 días por el mismo medio.
- El 6 de mayo se compra mercancía en Bodegas F por \$4500, pagando directamente en efectivo el 50%, y el restante mediante transferencia bancaria a 10 días.
- El 10 de mayo se vende mercancía por \$7600 a la empresa Nutri M, pagados así: el 40% por transferencia bancaria, y el 60 % restante se paga a 10 días.

- El 14 de mayo se reciben mediante transferencia bancaria \$50000 como pago final de Project Fitness.
- El 16 de mayo se paga \$150000 a la empresa Propiedad R el arriendo de la bodega; además, se paga el 50% restante adeudado a Bodegas F por \$2250.
- El 20 de mayo se recibe el pago del 70% restante de la venta hecha a Nutri M.
- El 25 de mayo se paga el 50% restante a Galería V por la compra de muebles para oficinas.

Traslado los anteriores movimientos financieros en una representación física del modelo presentado en la Figura 3- 2, son construidas las tablas de dimensiones y hechos para el manejo de hechos contables como se muestra en las Figura 3-3, Figura 3-4 y Figura 3- 5.

Figura 3-3: Dimensión Cuenta y sus niveles: a) Tipo Cuenta y b) Cuenta.

Nivel Tipo Cuenta		Nivel Cuenta		
Id_Cuenta	Tipo_Cuenta	Id_Cuenta	Cuenta	Id_Tipo_Cuenta
1	Activo	1	Cuenta x Pagar	2
2	Pasivo	2	Cuentas x Cobrar	1
3	Capital	3	Inventario	1
4	Ingresos	4	Caja	1
5	Costos	5	Banco	1
6	Gastos	6	Almacén	1
		7	Capital	3
		8	Gastos Administrativos	6

a) b)

Figura 3-4: Dimensión Cliente y su nivel Cliente.

Nivel Cliente	
Id_Cliente	Cliente
1	Galería V
2	Juan H
3	Bodegas F
4	Nutri M
5	Smart N
6	Propiedad R
7	Gym G
8	Project Fitness

Figura 3-6: Tabla de hechos (movimiento contable).

Id_Hecho_Contable	Fecha_Movimiento_Contable	Id_Cliente	Id_Cuenta_Origen	Id_Cuenta_Destino	Valor (\$)
1	01-may-18	5	7	4	200000
2	01-may-18	5	7	6	380000
10	04-may-18	8	6	4	79000
10	04-may-18	8	6	5	100000
10	04-may-18	8	6	3	60000
10	04-may-18	8	6	2	50000
3	05-may-18	1	1	3	2500
3	05-may-18	1	4	3	2500
5	06-may-18	3	4	6	2250
5	06-may-18	3	1	6	2250
6	10-may-18	4	6	5	3040
6	10-may-18	4	6	2	4560
10	14-may-18	8	2	5	50000
7	16-may-18	6	5	8	150000
5	16-may-18	3	5	1	2250
6	20-may-18	4	2	5	4560
3	25-may-18	1	5	1	2500

3.3 Agregaciones pictóricas y modelo de bodegas de datos enriquecido

La incorporación de pictogramas dentro de modelos de BD permite describir con precisión las áreas de estudio modeladas, diversas áreas de estudio como la geoinformática, medicina han enriquecido modelos conceptuales tradicionales de BD con objetos propios de su dominio como es mostrado en el capítulo 1, allí también se evidencia la carencia de elementos pictóricos que permitan describir entidades propias del álgebra lineal, esta área de estudio amalgamado con las BD permite desarrollar campos como la contabilidad matricial, análisis matricial de imágenes

En la Tabla 3-1 y Tabla 3-2 son presentados pictogramas(iconos) propuestos que representan elementos del álgebra matricial. Estos elementos corresponden a representaciones pictóricas para vectores y matrices, y para operaciones como suma, resta, multiplicación, división, transposición, inversión, etc.(Gamboa & Rodríguez, 2003)

Tabla 3-1: Tipos de datos (objetos) del álgebra matricial para el modelo conceptual de una BD.












Nombre	Pictograma	Observación
Matriz		
Vector		
Matriz virtual		Matriz calculada a partir de otros objetos (e.g., resultado de la suma de dos matrices). No se almacena físicamente en la BD.
Vector Virtual		Vector calculado a partir de otros objetos (e.g., resultado de la suma de dos vectores). No se almacena físicamente en la BD.

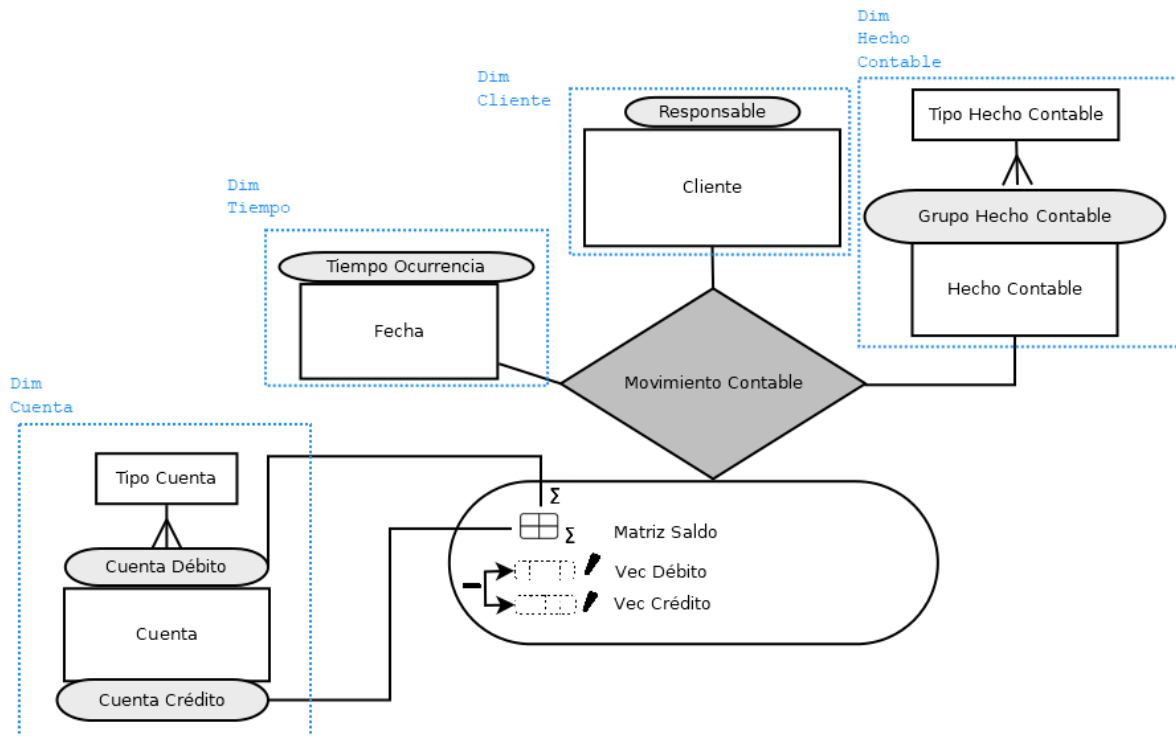
Tabla 3-2: Operaciones del álgebra matricial.

Nombre	Pictograma	Tipo de operación	Observación	Operandos
Suma		Binaria		Matriz + Matriz Matriz + Vector Vector + Vector Matriz + Escalar Vector + Escalar
Resta		Binaria		Análogo al anterior
Multiplicación		Binaria		Análogo al anterior
División		Binaria		Análogo al anterior
Media		Unaria	Promedio de los elementos de un objeto. Se puede aplicar por filas, por columnas, por diagonales o por el objeto completo, es decir, en el caso matricial podemos obtener el promedio de	Matriz, Vector

			cada fila o de una fila en particular.	
Sumatoria	Σ	Unaria	Sumatoria de los elementos de un objeto. Se aplica de forma análoga al anterior.	Matriz, Vector
Varianza	σ	Unaria	Varianza de los elementos de un objeto. Se aplica de forma análoga al anterior.	Matriz, Vector
Diagonal		Unaria	Extrae la diagonal de una matriz.	Matriz
Transpuesta		Unaria	Transpone el objeto.	Matriz, Vector
Inversa	-1	Unaria	Calcula la inversa de una matriz.	Matriz

- Los vectores y las matrices son atributos de una medida, de esta manera es posible operar con ellos.
- Las operaciones son divididas en dos tipos, binarias y unitarias, las operaciones binarias involucran más de un objeto dentro del cálculo realizado, las operaciones unitarias son operaciones que se realizan intra objetos, es decir el objeto se opera consigo mismo.
- La validez de la aplicación de una operación depende de la naturaleza del dominio del modelo, e.g., ¿tiene sentido en el ejercicio contable extraer la diagonal de la matriz de cuentas?, posiblemente no tendría ninguna aplicación práctica la extracción de tal diagonal.

El modelo presentado en la Figura 3- 2 es replanteado con los elementos pictóricos presentados en la Tabla 3-1 y Tabla 3-2, estos elementos combinados con la notación presentada en (Malinowski & Zimányi, 2008) permiten desarrollar un nuevo modelo para el manejo de hechos contables aplicando la teoría de contabilidad matricial con mayor simplicidad como se muestra en la Figura 3-7.

Figura 3-7: Modelo Conceptual BD hechos contables en forma matricial.

El modelo de la Figura 3-7 representa las medidas del hecho (Movimiento Contable) mediante una matriz y dos vectores (virtuales). Estos representan los valores de cada movimiento de un hecho contable mediante una matriz Cuentas asociada con la dimensión Cuenta (nótese que la matriz Cuentas tiene dos relaciones con la dimensión Cuenta). Los vectores virtuales Débitos y Créditos consolidan el estado de cada una de las cuentas, esto se logra al especificar al lado de la matriz los dos pictogramas Σ . Estos indican que es válido sumar sus filas y sus columnas.

Además, los vectores se pueden transponer (pictograma \diagup). Las operaciones permitidas entre los objetos se indican mediante flechas como se muestra en la Figura 3-7, e.g., la resta (pictograma $-$) de los dos vectores virtuales.

Cuando el número de operaciones entre objetos es alto (e.g., mayor que 4), se recomienda crear una tabla anexa para no recargar el modelo con flechas. Por ejemplo, en el modelo de la Figura 3-7 se puede quitar la flecha entre los vectores Débitos y Créditos y se reemplaza con la Tabla 3-3.

Tabla 3-3: Operaciones entre objetos.

Objeto 1	Objeto 2	Operación
Débitos	Créditos	Resta (-)

La Figura 3-8 presenta un ejemplo de la tabla de hechos “Movimientos Contables” del modelo propuesto en la Figura 3-7.

Figura 3-8: Tabla de Hechos con elementos del álgebra matricial.

Id_Cuenta (Crédito)

Id_Cuenta (Débito)

fecha	Id_Cliente	Id_Hecho_Contable	Matriz_Saldo	Vec_Credito	Vec_Debito																																																																																																																	
01-may	5	1	<table> <tr> <th></th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th>7</th> <th>8</th> </tr> <tr><th>1</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>2</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>3</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>4</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>5</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>6</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>7</th><td>0</td><td>0</td><td>0</td><td>200000</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>8</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>		1	2	3	4	5	6	7	8	1	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	7	0	0	0	200000	0	0	0	0	8	0	0	0	0	0	0	0	0	<table> <tr><th>1</th><td>0</td></tr> <tr><th>2</th><td>0</td></tr> <tr><th>3</th><td>0</td></tr> <tr><th>4</th><td>0</td></tr> <tr><th>5</th><td>0</td></tr> <tr><th>6</th><td>0</td></tr> <tr><th>7</th><td>200000</td></tr> <tr><th>8</th><td>0</td></tr> </table>	1	0	2	0	3	0	4	0	5	0	6	0	7	200000	8	0	<table> <tr><th>1</th><td>0</td></tr> <tr><th>2</th><td>0</td></tr> <tr><th>3</th><td>0</td></tr> <tr><th>4</th><td>200000</td></tr> <tr><th>5</th><td>0</td></tr> <tr><th>6</th><td>0</td></tr> <tr><th>7</th><td>0</td></tr> <tr><th>8</th><td>0</td></tr> </table>	1	0	2	0	3	0	4	200000	5	0	6	0	7	0	8	0
	1	2	3	4	5	6	7	8																																																																																																														
1	0	0	0	0	0	0	0	0																																																																																																														
2	0	0	0	0	0	0	0	0																																																																																																														
3	0	0	0	0	0	0	0	0																																																																																																														
4	0	0	0	0	0	0	0	0																																																																																																														
5	0	0	0	0	0	0	0	0																																																																																																														
6	0	0	0	0	0	0	0	0																																																																																																														
7	0	0	0	200000	0	0	0	0																																																																																																														
8	0	0	0	0	0	0	0	0																																																																																																														
1	0																																																																																																																					
2	0																																																																																																																					
3	0																																																																																																																					
4	0																																																																																																																					
5	0																																																																																																																					
6	0																																																																																																																					
7	200000																																																																																																																					
8	0																																																																																																																					
1	0																																																																																																																					
2	0																																																																																																																					
3	0																																																																																																																					
4	200000																																																																																																																					
5	0																																																																																																																					
6	0																																																																																																																					
7	0																																																																																																																					
8	0																																																																																																																					
01-may	5	2	<table> <tr> <th></th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th>7</th> <th>8</th> </tr> <tr><th>1</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>2</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>3</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>4</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>5</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>6</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>7</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>380000</td><td>0</td><td>0</td></tr> <tr><th>8</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>		1	2	3	4	5	6	7	8	1	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	7	0	0	0	0	0	380000	0	0	8	0	0	0	0	0	0	0	0	<table> <tr><th>1</th><td>0</td></tr> <tr><th>2</th><td>0</td></tr> <tr><th>3</th><td>0</td></tr> <tr><th>4</th><td>0</td></tr> <tr><th>5</th><td>0</td></tr> <tr><th>6</th><td>0</td></tr> <tr><th>7</th><td>380000</td></tr> <tr><th>8</th><td>0</td></tr> </table>	1	0	2	0	3	0	4	0	5	0	6	0	7	380000	8	0	<table> <tr><th>1</th><td>0</td></tr> <tr><th>2</th><td>0</td></tr> <tr><th>3</th><td>0</td></tr> <tr><th>4</th><td>0</td></tr> <tr><th>5</th><td>0</td></tr> <tr><th>6</th><td>380000</td></tr> <tr><th>7</th><td>0</td></tr> <tr><th>8</th><td>0</td></tr> </table>	1	0	2	0	3	0	4	0	5	0	6	380000	7	0	8	0
	1	2	3	4	5	6	7	8																																																																																																														
1	0	0	0	0	0	0	0	0																																																																																																														
2	0	0	0	0	0	0	0	0																																																																																																														
3	0	0	0	0	0	0	0	0																																																																																																														
4	0	0	0	0	0	0	0	0																																																																																																														
5	0	0	0	0	0	0	0	0																																																																																																														
6	0	0	0	0	0	0	0	0																																																																																																														
7	0	0	0	0	0	380000	0	0																																																																																																														
8	0	0	0	0	0	0	0	0																																																																																																														
1	0																																																																																																																					
2	0																																																																																																																					
3	0																																																																																																																					
4	0																																																																																																																					
5	0																																																																																																																					
6	0																																																																																																																					
7	380000																																																																																																																					
8	0																																																																																																																					
1	0																																																																																																																					
2	0																																																																																																																					
3	0																																																																																																																					
4	0																																																																																																																					
5	0																																																																																																																					
6	380000																																																																																																																					
7	0																																																																																																																					
8	0																																																																																																																					
04-may	8	10	<table> <tr> <th></th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th>7</th> <th>8</th> </tr> <tr><th>1</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>2</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>3</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>4</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>5</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>6</th><td>0</td><td>50000</td><td>60000</td><td>79000</td><td>100000</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>7</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>8</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>		1	2	3	4	5	6	7	8	1	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	6	0	50000	60000	79000	100000	0	0	0	7	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	<table> <tr><th>1</th><td>0</td></tr> <tr><th>2</th><td>0</td></tr> <tr><th>3</th><td>0</td></tr> <tr><th>4</th><td>0</td></tr> <tr><th>5</th><td>0</td></tr> <tr><th>6</th><td>289000</td></tr> <tr><th>7</th><td>0</td></tr> <tr><th>8</th><td>0</td></tr> </table>	1	0	2	0	3	0	4	0	5	0	6	289000	7	0	8	0	<table> <tr><th>1</th><td>0</td></tr> <tr><th>2</th><td>50000</td></tr> <tr><th>3</th><td>60000</td></tr> <tr><th>4</th><td>79000</td></tr> <tr><th>5</th><td>100000</td></tr> <tr><th>6</th><td>0</td></tr> <tr><th>7</th><td>0</td></tr> <tr><th>8</th><td>0</td></tr> </table>	1	0	2	50000	3	60000	4	79000	5	100000	6	0	7	0	8	0
	1	2	3	4	5	6	7	8																																																																																																														
1	0	0	0	0	0	0	0	0																																																																																																														
2	0	0	0	0	0	0	0	0																																																																																																														
3	0	0	0	0	0	0	0	0																																																																																																														
4	0	0	0	0	0	0	0	0																																																																																																														
5	0	0	0	0	0	0	0	0																																																																																																														
6	0	50000	60000	79000	100000	0	0	0																																																																																																														
7	0	0	0	0	0	0	0	0																																																																																																														
8	0	0	0	0	0	0	0	0																																																																																																														
1	0																																																																																																																					
2	0																																																																																																																					
3	0																																																																																																																					
4	0																																																																																																																					
5	0																																																																																																																					
6	289000																																																																																																																					
7	0																																																																																																																					
8	0																																																																																																																					
1	0																																																																																																																					
2	50000																																																																																																																					
3	60000																																																																																																																					
4	79000																																																																																																																					
5	100000																																																																																																																					
6	0																																																																																																																					
7	0																																																																																																																					
8	0																																																																																																																					

Los vectores virtuales se generan a partir de la suma de las filas y de las columnas de la matriz Cuentas como se ejemplifica en la Figura 3-9.

Figura 3-9: Ejemplo de generación de los vectores virtuales Débitos y Créditos.

La Matriz Matriz_Saldo del modelo de la Figura 3-7 se construye como se muestra en la Figura 3-10.

Figura 3-10: Matriz de cuentas Matriz_Saldo .

Cuenta (Débito)	Cuenta	1.Cuentas	2.Cuentas	3.Inventario	4.Caja	5.Banco	6.Almacén	7.Capital	8.Gastos
	(Débito)	por pagar	por cobrar						Administrativos
Cuenta (Crédito)									
1.Cuentas por pagar									
2.Cuentas por cobrar									
3.Inventario									
4.Caja									
5.Banco									
6.Almacén									
7.Capital									
8.Gastos administrativos									

Así, la Matriz_Saldo representa un movimiento contable donde un valor extraído de una cuenta de la fila i es depositado en una cuenta de la columna j . Nótese que la matriz posiblemente será dispersa (muchos elementos con valor 0). Por lo tanto, se requieren técnicas de compresión para optimizar su almacenamiento.

Ahora, considérese de nuevo la consulta d) planteada en el capítulo 3.1 Para resolverla en el nuevo modelo, ahora se deben sumar todos los vectores virtuales créditos y débitos de un mes, luego restar ambos vectores y compararlos contra la misma operación del mes anterior.

En general, todas las consultas de la capítulo 3.1 se simplifican. La Tabla 3-4 presenta un comparativo entre consultas hechas en el modelo convencional (Figura 3- 2) y en el matricial (Figura 3-7).

	iv) Restar los valores obtenidos en ii y iii	ambos vectores y el vector resultado tendrá el saldo final de cada cuenta.	este ordenamiento ni agrupamiento de cuentas.
--	--	--	---

Figura 3-11: Obtener el saldo de las cuentas que tuvieron movimientos el 4 y 6 de mayo de 2018 BD convencional.

i)

Id-Hecho_Contable	Fecha	Id_Cliente	Id_Cuenta_Origen	Id_Cuenta_Destino	Valor (\$)
1	01-may-18	5	7	4	200000
2	01-may-18	5	7	6	380000
10	04-may-18	8	6	4	79000
10	04-may-18	8	6	5	100000
10	04-may-18	8	6	3	60000
10	04-may-18	8	6	2	50000
3	05-may-18	1	1	3	2500
3	05-may-18	1	4	3	2500
5	06-may-18	3	4	6	2250
5	06-may-18	3	1	6	2250
6	10-may-18	4	6	5	3040
6	10-may-18	4	6	2	4560
10	14-may-18	8	2	5	50000
7	16-may-18	6	5	8	150000
5	16-may-18	3	5	1	2250
6	20-may-18	4	2	5	4560
3	25-may-18	1	5	1	2500

ii) y iii)

Id-Hecho_Contable	Fecha	Id_Cliente	Id_Cuenta_Origen	Id_Cuenta_Destino	Valor (\$)	Suma
10	04-may-18	8	6	4	79000	79000
10	04-may-18	8	6	5	100000	100000
10	04-may-18	8	6	3	60000	60000
10	04-may-18	8	6	2	50000	50000
5	06-may-18	3	4	6	2250	
5	06-may-18	3	1	6	2250	4500

iv) y v)

Id-Hecho_Contable	Fecha	Id_Cliente	Id_Cuenta_Origen	Id_Cuenta_Destino	Valor (\$)	Suma
10	04-may-18	8	6	4	79000	
10	04-may-18	8	6	5	100000	289000
10	04-may-18	8	6	3	60000	
10	04-may-18	8	6	2	50000	
5	06-may-18	3	4	6	2250	
5	06-may-18	3	1	6	2250	4500

vi)

Cuenta	Suma destino	Suma origen	Destino - Origen
1	0	2250	-2250
2	50000	0	50000
3	60000	0	60000
4	79000	2250	76750
5	100000	0	100000
6	4500	289000	-284500

Figura 3-12: Obtener el saldo de las cuentas que tuvieron movimientos el 4 y 6 de mayo de 2018 BD Matricial.

i)



Día	Id_Cliente	Id_Hecho	Cuentas	Créditos	Débitos
04-may	8	10	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
			1 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0
			2 0 0 0 0 0 0 0	2 0 0 0 0 0 0 0	2 0 0 0 0 0 0 0
			3 0 0 0 0 0 0 0	3 0 0 0 0 0 0 0	3 0 0 0 0 0 0 0
			4 0 0 0 0 0 0 0	4 0 0 0 0 0 0 0	4 0 0 0 0 0 0 0
			5 0 0 0 0 0 0 0	5 0 0 0 0 0 0 0	5 0 0 0 0 0 0 0
			6 50000 60000 79000 100000 0 0 0	6 289000 0 0 0 0 0 0	6 0 0 0 0 0 0 0
			7 0 0 0 0 0 0 0	7 0 0 0 0 0 0 0	7 0 0 0 0 0 0 0
			8 0 0 0 0 0 0 0	8 0 0 0 0 0 0 0	8 0 0 0 0 0 0 0
06-may	3	5	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
			1 0 0 0 0 2250 0 0	1 2250 0 0 0 0 0 0	1 0 0 0 0 0 0 0
			2 0 0 0 0 0 0 0	2 0 0 0 0 0 0 0	2 0 0 0 0 0 0 0
			3 0 0 0 0 0 0 0	3 0 0 0 0 0 0 0	3 0 0 0 0 0 0 0
			4 0 0 0 0 0 2250 0	4 2250 0 0 0 0 0 0	4 0 0 0 0 0 0 0
			5 0 0 0 0 0 0 0	5 0 0 0 0 0 0 0	5 0 0 0 0 0 0 0
			6 0 0 0 0 0 0 0	6 0 0 0 0 0 0 0	6 0 0 0 0 0 0 0
			7 0 0 0 0 0 0 0	7 0 0 0 0 0 0 0	7 0 0 0 0 0 0 0
			8 0 0 0 0 0 0 0	8 0 0 0 0 0 0 0	8 0 0 0 0 0 0 0

ii)

1	2	3	4	5	6	7	8
1 0	0	0	0	0	0	0	0
2 0	0	0	0	0	0	0	0
3 0	0	0	0	0	0	0	0
4 0	0	0	0	0	0	0	0
5 0	0	0	0	0	0	0	0
6 50000	60000	79000	100000	0	0	0	0
7 0	0	0	0	0	0	0	0
8 0	0	0	0	0	0	0	0

1	2	3	4	5	6	7	8
1 0	0	0	0	0	2250	0	0
2 0	0	0	0	0	0	0	0
3 0	0	0	0	0	0	0	0
4 0	0	0	0	0	2250	0	0
5 0	0	0	0	0	0	0	0
6 0	0	0	0	0	0	0	0
7 0	0	0	0	0	0	0	0
8 0	0	0	0	0	0	0	0

1	2	3	4	5	6	7	8
1 0	0	0	0	0	2250	0	0
2 0	0	0	0	0	0	0	0
3 0	0	0	0	0	0	0	0
4 0	0	0	0	0	2250	0	0
5 0	0	0	0	0	0	0	0
6 50000	60000	79000	100000	0	0	0	0
7 0	0	0	0	0	0	0	0
8 0	0	0	0	0	0	0	0

iii) y iv)



Créditos	Débitos
1 2250	1 0
2 0	2 50000
3 0	3 60000
4 2250	4 79000
5 0	5 100000
6 289000	6 4500
7 0	7 0
8 0	8 0

Saldo
1 -2250
2 50000
3 60000
4 76750
5 100000
6 248500
7 0
8 0

La tabla de hechos (Figura 3-8) se puede gestionar mediante Cubos OLAP(D. Orlando et al., 2007), el cual permite seleccionar dimensiones específicas y disminuir el nivel de detalle al agregar las medidas (los valores). Por ejemplo, si se quita la dimensión Hecho Contable de la Tabla de hechos de la Figura 3-8, se generan las matrices y vectores mostrados en la Figura 3-13, donde los dos primeros registros de la tabla de hechos (Figura 3-8) se fusionan en un registro.

Figura 3-13: Iteración sobre el Hecho.

Fecha	Cliente	Matriz								Créditos	Débitos
01-may	5	1	2	3	4	5	6	7	8		
		1	0	0	0	0	0	0	0	1	0
		2	0	0	0	0	0	0	0	2	0
		3	0	0	0	0	0	0	0	3	0
		4	0	0	0	0	0	0	0	4	200000
		5	0	0	0	0	0	0	0	5	0
		6	0	0	0	0	0	0	0	6	380000
		7	0	0	0	200000	0	380000	0	7	0
		8	0	0	0	0	0	0	0	8	0

Así, es posible continuar agregando las medidas, e.g., al quitar la dimensión Cliente se pueden obtener las medidas agregadas a nivel de día. Adicionalmente, si los días se agrupan por mes, el resultado se muestra en la Figura 3-14. De esta manera, la comparación entre periodos se facilita, además los débitos y los créditos consolidados quedan implícitos en la tabla de hechos.

Figura 3-14: Tabla de hechos agrupados por Mes.

Mes	Matriz									Crédito		Débito	
Mayo		1	2	3	4	5	6	7	8				
1				2500			2250			1	5000	1	4500
2						54560				2	54560	2	54560
3										3	0	3	5000
4				2500			2250			4	4750	4	279000
5	4500								150000	5	154500	5	157600
6			54560		79000	103040				6	236600	6	384500
7					200000		380000			7	580000	7	0
8										8	0	8	150000

4. Implementación modelo bodega de datos con variables matriciales para el manejo de hechos contables.

El uso de objetos matriciales en los modelos de BD simplifica la construcción y la explicación de modelos lógicos donde los objetos del álgebra matricial desempeñan un rol importante como en la contabilidad matricial, e.g., para la gestión de cuentas pertenecientes a un estado financiero.

Conjuntamente, los modelos conceptuales pueden ser fácilmente llevados a modelos lógicos implementados en sistemas de gestión de BD ya que estos incluyen en sus tipos de datos, arreglos (*array*) de n dimensiones (Steinbach & Copping, 2019; TERADATA, 2018), elemento principal para la construcción de vectores y matrices.

4.1 Entorno de desarrollo

Para la transformación y desarrollo de los modelos conceptuales en modelos físicos se utilizó el sistema de datawarehouse Apache Hive (Apache Hive, 2011), que facilita el almacenamiento y el manejo de grandes volúmenes de datos usando el HDFS (Hadoop Distributed File System) de Hadoop,

Apache Hive proporciona un lenguaje de consultas basado en SQL llamado HiveQL el cual lee y convierte las consultas al modelo de computación paralela MapReduce que genera estrategias de distribución de los datos por nodos.

Esta infraestructura es proporcionada por la máquina virtual Cloudera QuickStart en la versión de Hadoop y Hive mostrada a continuación:

- Hadoop: 2.6.0-cdh5.13.0
- Hive: 1.1.0-cdh5.13.0

Además, la máquina virtual se aloja sobre un sistema Windows 10 de 64 bits con 16 Gigabytes en memoria RAM y un procesador Intel Core i7, usando 4 Gigabytes en memoria RAM y un procesador.

4.2 Implementación de modelos físicos a partir de modelos conceptuales

Los modelos conceptuales presentados en la Figura 3- 2 y Figura 3-7 son implementados en este capítulo con el objetivo de contrastar los tipos de consultas que se pueden plantear para resolver las preguntas presentadas en el Capítulo 3 y desarrolladas en la Tabla 3-4.

Para el desarrollo del modelo conceptual tradicional de la Figura 3- 2 se crean las Dimensiones

- Tipo_Cuenta
- Cuenta
- Cliente
- Tipo_Hecho_Contable
- Hecho_Contable.

Y la tabla de hechos FCT_Movimiento_Contable, la cual cuenta con la estructura mostrada en la Tabla 4-1.

Tabla 4-1: Estructura tabla de hechos modelo convencional.

Nombre columna	Tipo de dato
Id_Movimiento_Contable	INTEGER (entero)
Fecha_Movimiento_Contable	DATE
Id_Cliente	INTEGER (entero)
Id_Cuenta_Origen	INTEGER (entero)
Id_Cuenta_Destino	INTEGER (entero)
Saldo	DOUBLE

EL modelo conceptual matricial de la Figura 3-7 usa las mismas dimensiones, pero la tabla de hechos se replantea con el objetivo de incluir en esta las nuevas variables de tipo arreglo. La tabla FCT_Movimiento_Contable_MAT sigue la estructura presentada en la Tabla 4-2.

Tabla 4-2: Estructura tabla de hechos modelo matricial.

Nombre columna	Tipo de dato
Id_Movimiento_Contable_Mat	INTEGER (entero)
Fecha_Movimiento_Contable_Mat	DATE
Id_Cliente_Mat	INTEGER (entero)
Id_Hecho_Contable	INTEGER (entero)
Matriz_Saldo	Array(Array(DOUBLE))

Para la construcción de la matriz (matriz saldo) es necesario implementar un arreglo de dos dimensiones, el tipo de dato de cada entrada i, j del arreglo de arreglos o matriz será del tipo real (DOUBLE).

La diferencia entre las dos tablas de hechos radica en la relación que tienen las métricas con las dimensiones que construyen la matriz, en la tabla de hechos convencional se da una doble relación con la dimensión cuenta, mientras que en la tabla de hechos matricial la doble relación con la dimensión cuenta está representada en las columnas y filas que componen la matriz, como se observa en el modelo de la Figura 3-7.

Algunos sistemas de gestión de BD como Teradata y ORACLE implementan diferentes tipos de funciones para el manejo de tipos de datos especiales como arreglos (TERADATA, n.d.), en Hive el desarrollo de funciones propias para el manejo de tipos de datos especiales como arreglos aún no se encuentran muy desarrolladas, pero este sistema permite definir funciones propias del Usuario (*UDFs- User-Defined Functions*) sobre Java, las cuales se pueden invocar desde JDBC (API para que permite la ejecución de operaciones sobre una base de datos desde JAVA) .

El prototipo se implementa en Hive por la facilidad de creación de las funciones propias del usuario sobre Java y la invocación de las mismas desde JDBC, ya que TERADATA no

proporciona las funciones suficientes para el desarrollo pleno del ejercicio y la creación de funciones propias del usuario, en este sistema se requieren configuraciones especiales de librerías ya que las funciones propias del usuario se deben definir en C o C++; Hive al estar implementado en Java permite una transparencia entre objetos nativos de Hive y objetos JAVA.

4.2.1 Funciones Definidas por el usuario para el manejo de arreglos

Para la implementación del prototipo se desarrollaron varias funciones que permiten operar arreglos, como se muestra en la Tabla 3-2. La Tabla 4-3 muestra las funciones programadas, los tipos de datos de entrada y los tipos de datos que retorna

Tabla 4-3: Funciones definidas por el usuario. ²

Nombre Función	Operación	Tipo dato entrada	Tipo Dato salida	Observación
MAT_SUB_VEC	Resta	vectores	vectores	Recibe dos vectores del mismo tamaño y realiza la resta de cada una de las entradas retornando un nuevo vector
MAT_SUM_COL	Suma	Matriz	Vector	Recibe una matriz y realiza la suma de todos los elementos de una misma columna, retornando un vector con las diferentes sumatorias de cada columna como se muestra en la Figura 3-9
MAT_SUM_ROW	Suma	Matriz	Vector	Recibe una matriz y realiza la suma de todos los elementos de una misma fila, retornando un vector con las diferentes sumatorias de cada fila como se muestra en la Figura 3-9
MAT_SUM_ALL	Suma	Matriz	Matriz	Recibe n de igual tamaño matrices y devuelve una única matriz de tamaño n

² El anexo B contiene los scripts de creación de cada una de las funciones mencionadas en la Tabla 4-3.

				con la sumaria de todas las diferentes entradas i j
--	--	--	--	---

4.3 Consultas Modelo lógico convencional y matricial

Los enriquecimientos pictóricos sobre modelos conceptuales de BD acompañados de desarrollos en sistemas de gestión de BD facilitan el planteamiento de consultas sobre los modelos lógicos que estos representan, a continuación, se presentan las consultas SQL planteadas en la Tabla 3-4, las consultas se proponen sobre un modelo BD convencional y sobre un modelo BD matricial para el manejo de hechos contables.

4.3.1 Consulta 1

¿Obtener el saldo de las cuentas que tuvieron movimientos el 4 y 6 de mayo de 2018?

- Convencional:

```

SELECT
    COALESCE(origen.id_cuenta_origen,destino.id_cuenta_destino)
Cuenta,
    (COALESCE(destino.sum_destino,0) -
COALESCE(origen.sum_origen,0)) Saldo_final
FROM
    (SELECT
        id_cuenta_origen,
        SUM(valor) sum_origen
    FROM fct_movimiento_contable
    WHERE fecha_movimiento_contable = '2018-05-04'
or fecha_movimiento_contable = '2018-05-06'
    GROUP BY id_cuenta_origen) origen
FULL JOIN
    (SELECT
        id_cuenta_destino,
        SUM(valor) sum_destino
    FROM fct_movimiento_contable
    WHERE fecha_movimiento_contable = '2018-05-04' or
fecha_movimiento_contable = '2018-05-06'
    GROUP BY id_cuenta_destino) destino
ON (destino.id_cuenta_destino=origen.id_cuenta_origen);

```

- Matricial:

```

SELECT

```

```

MAT_SUB_VEC
(
    MAT_SUM_COL(MAT_SUM_ALL(matriz_saldo)),
    MAT_SUM_ROW(MAT_SUM_ALL(matriz_saldo))
)
FROM fct_movimiento_contable_mat
WHERE fecha_movimiento_contable = '2018-05-04' OR
fecha_movimiento_contable = '2018-05-06';

```

Comparando las consultas de un modelo convencional y del modelo matricial, se observa la simplicidad de las segundas usando funciones definidas por el usuario, estas funciones enriquecen el lenguaje.

4.3.2 Consulta 2

¿Cuál es el estado final de la cuenta por pagar asociada con el cliente 1 el mes de Mayo?

- Convencional:

```

SELECT
    COALESCE(destino.id_cliente, origen.id_cliente),
    (COALESCE(origen.Saldo_origen) - COALESCE(destino.Saldo_Destino))
saldo_final
FROM
(
    SELECT
        fmc.id_cliente AS id_cliente,
        SUM(fmc.valor) AS Saldo_Origen
    FROM fct_movimiento_contable fmc
    INNER JOIN
    Cuenta cu
    ON(fmc.id_cuenta_origen = cu.Id_Cuenta)
    WHERE fmc.id_cliente =1
    AND fmc.fecha_movimiento_contable
    BETWEEN '2018-05-01' AND '2018-05-31'
    AND cu.id_cuenta=1
    AND fmc.id_cuenta_origen =1
    GROUP BY fmc.id_cliente
) origen
INNER JOIN
(
    SELECT
        fmc.id_cliente AS id_cliente,
        SUM(fmc.valor) AS Saldo_Destino
    FROM fct_movimiento_contable fmc
    INNER JOIN
    Cuenta cu

```

```

    ON (fmc.Id_cuenta_destino = cu.Id_Cuenta)
    WHERE fmc.id_cliente =1
    AND fmc.fecha_movimiento_contable
    BETWEEN '2018-05-01' AND '2018-05-31'
    AND fmc.id_cuenta_destino=1
    GROUP BY fmc.id_cliente
  ) destino
ON (destino.id_cliente=origen.id_cliente)

```

Mediante funciones especiales como GROUPING SETS (Jain & Camacho, 2017) incluidas dentro de los sistemas de gestión de BD la consulta planteada se puede simplificar, aunque este planteamiento requiere un conocimiento más amplio de SQL ya que estas funciones de agrupamiento permiten crear más de un grupo en el mismo conjunto de registros, como se muestra a continuación.

- Convencional GROUPING SETS:

```

SELECT
  MAX(t1.origen)-MAX(t1.destino) saldo_final
FROM
  (
    SELECT
      CASE WHEN id_cuenta_origen = 1 THEN SUM(valor) ELSE 0 END AS
origen,
      CASE WHEN id_cuenta_destino =1 THEN SUM(valor) ELSE 0 END AS
destino
    FROM fct_movimiento_contable
    WHERE id_cuenta_origen = 1 or id_cuenta_destino = 1
    GROUP BY id_cuenta_origen,id_cuenta_destino
    GROUPING SETS ((id_cuenta_origen),(id_cuenta_destino))
  ) t1

```

- Matricial:

```

SELECT
  t2.id_cliente,
  t2.cuenta,
  t2.valor
  FROM
    (
      SELECT
        t1.id_cliente id_cliente,
        i cuenta,
        c valor

```

```

FROM
(
    SELECT
        fctm.id_cliente,

        MAT_SUB_VEC(MAT_SUM_COL(MAT_SUM_ALL(matriz_saldo)),MAT_SUM_ROW(MAT_
SUM_ALL(matriz_saldo)))mat_saldo
        FROM fct_movimiento_contable_mat fctm
        WHERE id_cliente = 1 AND fecha_movimiento_contable
        BETWEEN '2018-05-01' AND '2018-05-31'
        GROUP BY fctm.id_cliente

    ) t1
    LATERAL VIEW posexplode(t1.mat_saldo) v1 AS i, c
    )
    t2
INNER JOIN
Cuenta cu
ON(t2.cuenta = cu.Id_Cuenta)
WHERE cu.Id_Cuenta=1;

```

Los tres tipos de consultas planteadas parecen extensas en su composición, aunque la consulta en forma matricial es suficiente a partir del segundo FROM anidado, las demás subconsultas solo son para extraer del arreglo resultante los valores e índices, que posteriormente son unidos con la dimensión cuenta.

4.3.3 Consulta 3

¿Cuál es el estado final de todas las cuentas el día 30 del mes anterior?

- Convencional:

```

SELECT
    COALESCE(origen.id_cuenta_origen,destino.id_cuenta_destino) Cuenta,
    (COALESCE(destino.sum_destino,0)- COALESCE(origen.sum_origen,0))
Saldo_final
FROM
(
    SELECT
        id_cuenta_origen,
        SUM(valor) sum_origen
    FROM
        fct_movimiento_contable
    WHERE fecha_movimiento_contable
    BETWEEN '2018-05-01' AND '2018-05-31'
    GROUP BY id_cuenta_origen
) origen
FULL JOIN
(

```



```

SELECT
    id_cuenta_destino,
    SUM(valor) sum_destino
FROM
    fct_movimiento_contable
WHERE fecha_movimiento_contable
    BETWEEN '2018-05-01' AND '2018-05-31'
GROUP BY id_cuenta_destino
) destino
ON (destino.id_cuenta_destino=origen.id_cuenta_origen);

```

Matricial:

```

SELECT

    MAT_SUB_VEC(MAT_SUM_COL(MAT_SUM_ALL(matriz_saldo)),

    MAT_SUM_ROW(MAT_SUM_ALL(matriz_saldo)))

FROM fct_movimiento_contable_mat

WHERE fecha_movimiento_contable

    BETWEEN '2018-05-01' AND '2018-05-31';

```

Las consultas planteadas sobre el modelo lógico matricial reflejan simplicidad en su construcción, además de presentar un mejor rendimiento sobre las consultas realizadas sobre el modelo lógico convencional como se muestra en la Tabla 4-4 donde se contrastan los tiempos de ejecución en ambos modelos.

Tabla 4-4: Tiempo ejecución consultas.

Consulta	Tiempo ejecución convencional	Tiempo ejecución Matricial
1	2 minutos 9 segundos	39.9 segundos
2 ³	2 minutos 22 segundos	1 minuto 6 segundos
3	2 minutos 55 segundos	47.26 segundos

³ La consulta Convencional planteada con GROUPING SET presento un tiempo de 58.44 segundos.

5. Conclusiones y recomendaciones

5.1 Conclusiones

Los enriquecimientos pictóricos sobre modelos conceptuales de BD permiten el desarrollo de los diferentes campos de estudio donde estas mejoras son aplicadas, permitiendo formalizar el manejo de tipos de datos complejos (i.e., datos no escalares) o habilitando la fácil construcción y explicación de los modelos conceptuales, además facilitan la aplicación de estos en modelos lógicos.

Los objetos agregados sobre los modelos deben ser entidades aportantes que tengan la capacidad de describir gráficamente aspectos importantes de los diferentes dominios, se debe considerar que su estructura pueda ser plasmable e identificable fácilmente mediante iconos o pictogramas, no tiene sentido agregar iconos que no se asemejen a la entidad que pretenden representar, por el contrario, harían engorroso el proceso de entendimiento y planteamiento de los modelos.

En los últimos años diversos autores han realizado valiosos aportes en campos como la geoinformática, la biología o la medicina, incorporando iconos que permiten representar el tiempo, el espacio u objetos como trayectorias, ambulancias o puntos de interés, lo que facilita la lectura y la construcción de un modelo conceptual.

Se identificó una clara carencia en disciplinas como la contabilidad, el álgebra matricial o la unión de estas dos disciplinas, diversos campos de estudio (e.g., análisis de imágenes, biología, contabilidad) usan componentes del álgebra matricial para mejora su estructura, por lo tanto, un enriquecimiento en esta área permite transversalmente enriquecer más de un área del conocimiento.

Las adiciones pictóricas propuestas en este trabajo consideran el álgebra matricial aplicada (e.g., contabilidad matricial), al incorporar sobre modelos conceptuales

pictogramas de elementos esenciales de esta disciplina como matrices, vectores y operaciones, esto a su vez permite formular modelos conceptuales robustos y entendibles, además de modelos matemáticos con cálculos complejos que son simplificados mediante la operación de las matrices y vectores y toda la teoría que se encuentra detrás de esta disciplina; este contraste es identificable en la comparación hecha en el Capítulo 3 sobre un modelo de BD convencional para el manejo de hechos contables y un modelo de BD matricial con los enriquecimientos propuestos, donde al plasmarse objetos como matrices y vectores es más entendible para el lector que tema se está trabajando dentro del modelo conceptual, es importante considerar las restricciones naturales que contiene operar objetos propios del álgebra matricial, como la multiplicación de una matriz por un vector, donde el número de columnas de la matriz debe ser igual al número de filas del vector para que la operación sea válida y tenga sentido; además de esta restricción surgen muchas más restricciones propias del álgebra matricial como la invariabilidad de matrices entre otras.

Los modelos lógicos (i.e., en tablas de hechos y dimensiones) facilitan las diferentes consultas sobre las BD, acortando los pasos y ordenándolos de forma intuitiva como se muestra en la Tabla 3-4, el Capítulo 4 desarrolla dentro de un sistema de gestión de BD como Hive algunos de los conceptos propuestos en la Capítulo 3; usando tipos de datos nativos de Hive e implementando funciones definidas por el usuario es posible plasmar la simplicidad de las consultas de la Tabla 3-4 en lenguaje SQL, donde es evidente como una misma consulta se puede plantear en menos pasos.

Mediante funciones definidas por el usuario desarrolladas para el manejo de tipo de datos arreglo, es posible generar consultas simples e intuitivas que presentan **mejor rendimiento** que consultas tradicionales sobre un modelo lógico convencional de BD como muestra la Tabla 4-4, donde se contrastan los tiempos de ejecución de consultas sobre ambos modelos.

5.2 Recomendaciones

Las restricciones dentro de las operaciones del álgebra matricial son un tema crucial en el planteamiento de trabajos de este tipo; por lo tanto, su implementación debe ser más robusta dentro de las adiciones pictóricas propuestas a los modelos de BD.

El desarrollo de funciones definidas por el usuario es un tema de gran relevancia que SE debe impulsar y mejorar, ya que estas operaciones permiten enriquecer el lenguaje SQL y plantear de forma intuitiva las consultas necesarias para potenciar las diferentes áreas del conocimiento; en el desarrollo de estas funciones se deben considerar aspectos como el paralelismo o el rendimiento de las consultas cuando se trate de grandes volúmenes de datos (i.e., paradigma o programación bajo el marco MapReduce).

A. Anexo: Tablas modelo lógico

En el capítulo 4 de este trabajo se presentó la implementación de un modelo conceptual a un modelo lógico, en este anexo son presentadas las consultas DDL (Data Definition Language) utilizadas en la creación de la base de datos y tablas, además de las consultas DML (Data Manipulation Language) usadas para poblar las diferentes tablas de la base de datos.

A.1 Tablas de dimensiones

```
CREATE TABLE IF NOT EXISTS Tipo_Cuenta
(Id_Tipo_Cuenta INT COMMENT 'id del tipo de cuenta',
Tipo_Cuenta VARCHAR(50) COMMENT 'nombre del tipo cuenta'
)
COMMENT 'tabla tipo cuenta padre de la tabla cuenta n2';
```

```
CREATE TABLE IF NOT EXISTS Cuenta
(Id_Cuenta INT COMMENT 'id de la cuenta',
Cuenta_Desc VARCHAR(50) COMMENT 'nombre del tipo cuenta',
Id_Tipo_Cuenta INT COMMENT 'referencia el campo Id_tipo_cuenta de la
tabla Tipo_cuenta'
)
COMMENT 'tabla cuenta hija de la tabla tipo cuenta n1';
```

```

CREATE TABLE IF NOT EXISTS contabilidad.Cliente
(Id_Cliente INT COMMENT 'id del cliente',
Cliente_Desc VARCHAR(50) COMMENT 'nombre del cliente'
)
COMMENT 'tabla de clientes n1';

```

```

CREATE TABLE IF NOT EXISTS Tipo_Hecho_Contable
(Id_Tipo_Hecho_Contable INT COMMENT 'id del Tipo_Hecho_Contable',
Tipo_Hecho_Contable_Desc VARCHAR(100) COMMENT 'nombre
Tipo_Hecho_Contable'
)
COMMENT 'tabla Tipo_Hecho_Contable n2';

```

```

CREATE TABLE IF NOT EXISTS Hecho_Contable
(Id_Hecho_Contable INT COMMENT 'id del Hecho_Contable',
Id_Tipo_Hecho_Contable INT COMMENT 'id del Tipo_Hecho_Contable
referencia la tabla tipo_hecho_contable'
)
COMMENT 'tabla Hecho_Contable n1';

```

A.2 Tablas de Hechos y vista

```

CREATE TABLE IF NOT EXISTS FCT_Movimiento_Contable
(Id_Movimiento_Contable INT COMMENT 'id del Movimiento_Contable',
Fecha_Movimiento_Contable DATE COMMENT 'fecha en que ocurrio el
movimiento contable',
Id_Cliente INT COMMENT 'id del cliente referencia el nivel cliente',
Id_Cuenta_Origen INT COMMENT 'id de la cuenta origen referencia el nivel
cuenta',
Id_Cuenta_Destino INT COMMENT 'id de la cuenta Destino referencia el
nivel cuenta',
Saldo DOUBLE COMMENT 'saldo del movimiento contable'

```

)

COMMENT 'tabla de hechos principal ';

CREATE TABLE IF NOT EXISTS FCT_Movimiento_Contable_MAT

(Id_Movimiento_Contable_Mat **INT COMMENT** 'id del Movimiento_Contable',

Fecha_Movimiento_Contable_Mat **DATE COMMENT** 'fecha en que ocurrio el movimiento contable',

Id_Cliente_Mat **INT COMMENT** 'id del cliente referencia el nivel cliente',

Id_Hecho_Contable **INT COMMENT** 'id del hecho contable',

Matriz_Saldo **ARRAY<ARRAY<DOUBLE>>** **COMMENT** 'MATriz de saldos donde las entradas son las cuentas de la tabla cuenta'

)

COMMENT 'tabla de hechos principal en forma matrical ';

CREATE VIEW VE_Movimiento_Contable_MAT **AS**

SELECT

Id_Movimiento_Contable_Mat,

Fecha_Movimiento_Contable_Mat,

Id_Cliente_Mat,

Id_Hecho_Contable,

Matriz_Saldo,

MAT_SUM_ROW(Matriz_Saldo) **AS** Credito_Cuenta,

MAT_SUM_COL(Matriz_Saldo) **AS** Debito_Cuenta

FROM contabilidad.FCT_Movimiento_Contable_MAT

;

B. Anexo: Funciones definidas por el usuario Hive-Java

Las funciones definidas por el usuario (UDF's) permiten enriquecer la sintaxis de los sistemas de gestión de BD, Hive permite definir este tipo de funciones mediante scripts de Java, los cuales son posteriormente almacenados en .JAR e invocados desde el JDBC como una función temporal o permanente.

B.1 Función MAT_SUM_ALL

Función de agregación, suma todas las matrices de los registros enviados, retornado una sola matriz.

```
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.apache.hadoop.hive.ql.exec.Description;
import org.apache.hadoop.hive.ql.exec.UDAF;
import org.apache.hadoop.hive.ql.exec.UDAFEvaluator;
import org.apache.hadoop.hive.ql.metadata.HiveException;
import java.util.ArrayList;
import java.util.Collection;
import java.util.Collections;
import java.util.List;
import org.apache.hadoop.io.IntWritable;

public class MAT_SUM_ALL extends UDAF
{
    static final Log LOG =
LogFactory.getLog(UDAF_ARRAY.class.getName());
    public static class MeanUDAFEvaluator implements UDAFEvaluator
    {
        ArrayList<ArrayList<Double>> col = null;
        int nombreConstante;

        public MeanUDAFEvaluator()
        {
            super();
            init();
        }
    }
}
```

```

    }

    //Metodo init, inicialización del array que se retorna
    public void init()
    {
        LOG.debug("Initialize evaluator");
        col = new ArrayList<ArrayList<Double>>();
    }

    //Metodo iterate, el arreglo que se retornó se llena con valores
    0 para el manejo de null, luego se suman los valores que entran con los
    valores que se deben retornar
    public boolean iterate(ArrayList<ArrayList<Double>> value)
    throws HiveException
    {
        LOG.debug("Iterating over each value for aggregation");
        if (col == null)
            throw new HiveException("Item is not initialized");

        int n = value.size();
        int i,j = 0;
        double aux = 0;

        for (i = 0; i < n; ++i)
        {
            try {
                col.get(i);
            } catch (IndexOutOfBoundsException e)
            {
                col.add(i, new
ArrayList<Double>(Collections.nCopies(n, 0.0)));
            }

            for (j = 0; j < n; ++j)
            {
                // añadir elementos
                aux = col.get(i).get(j) + value.get(i).get(j) ;
                col.get(i).set(j,aux);
                aux =0;
            }
        }
        return true;
    }

    //Metodo terminatePartial, retorno parcial del arreglo
    public ArrayList<ArrayList<Double>> terminatePartial()
    {
        LOG.debug("Return partially aggregated results");
        return col;
    }

    //Metodo merge, unión de los valores parciales
    public boolean merge(ArrayList<ArrayList<Double>> other) throws
HiveException
    {
        LOG.debug("merging by combining partial aggregation");
    }

```

```

        if(other == null)
        {
            return true;
        }

        int n = other.size();
        int i,j = 0;
        double aux = 0;

        for (i = 0; i < n; ++i)
        {
            try {
                col.get(i);
            } catch (IndexOutOfBoundsException e)
            {
                col.add(i, new
ArrayList<Double>(Collections.nCopies(n, 0.0))); //Add something to fill
this position.
            }

            for (j = 0; j < n; ++j)
            {
                aux = col.get(i).get(j) + other.get(i).get(j) ;
                col.get(i).set(j,aux);
                aux =0;
            }

        }
        return true;
    }
    //Metod terminate, retorno final
    public ArrayList<ArrayList<Double>> terminate()
    {
        LOG.debug("At the end of last record of the group -
returning final result");
        return col;
    }
}
}

```

B.2 Función MAT_SUB_VEC

Función de registro, resta de vectores, la operación se realiza individual a cada registro enviado.

```

import java.util.ArrayList;
import java.util.Collection;
import java.util.List;
import org.apache.hadoop.hive.ql.exec.UDF;

```

```

import org.apache.hadoop.io.IntWritable;

public class MAT_SUB_VEC extends UDF
{
    public ArrayList<Double > evaluate(ArrayList<Double > debito,
    ArrayList<Double > credito)
    {
        ArrayList<Double > result = new ArrayList<Double >();

        if (debito == null || debito.size() < 1 || credito == null ||
credito.size() < 1)
        {
            result.add(0.0);
            return result;
        }
        int i = 0;
        double sub=0.0;
        int n = debito.size();

        ArrayList<Double > substra =new ArrayList<Double >() ;
        for (i = 0; i < n; ++i)
        {

            // Añadir elemento
            sub = debito.get(i) - credito.get(i);
            substra.add(i, sub);
            sub = 0;
        }
        return substra;
    }
}

```

B.3 Función MAT_SUM_COL

Función de registro, suma columnas de una matriz retornando un vector de cada sumatoria, la operación se realiza individual a cada registro enviado.

```

import java.util.ArrayList;
import java.util.Collection;
import java.util.List;
import org.apache.hadoop.hive ql.exec.UDF;
import org.apache.hadoop.io.IntWritable;

public class MAT_SUM_COL extends UDF
{

```

```

    public ArrayList<Double > evaluate(ArrayList<ArrayList<Double >>
list)
    {

        ArrayList<Double > result = new ArrayList<Double >();

        if (list == null || list.size() < 1)
        {
            result.add(0.0);
            return result;
        }

        int i,j = 0;
        double sum=0.0;
        int n = list.size();
        int n2=list.get(0).size();

        ArrayList<Double > subList = new ArrayList<Double >() ;

        for (i = 0; i < n2; ++i)
        {
            for (j = 0; j < n; ++j)
            {

                // Add the element
                sum = sum + list.get(j).get(i);

            }
            subList.add(i, sum);

            // Reset the sum
            sum = 0;
        }

        return subList;
    }
}

```

B.4 Función MAT_SUM_ROW

Función de registro, suma filas de una matriz retornando un vector de cada sumatoria, la operación se realiza individual a cada registro enviado.

```

import java.util.ArrayList;
import java.util.Collection;
import java.util.List;
import org.apache.hadoop.hive.ql.exec.UDF;
import org.apache.hadoop.io.IntWritable;

public class MAT_SUM_ROW extends UDF
{

    public ArrayList<Double > evaluate(ArrayList<ArrayList<Double >>
list)
    {

        ArrayList<Double > result = new ArrayList<Double >();

        if (list == null || list.size() < 1)
        {
            result.add(0.0);
            return result;
        }

        int i,j = 0;
        double sum=0.0;
        int n = list.size();
        int n2=list.get(0).size();

        ArrayList<Double > subList = new ArrayList<Double >() ;

        for (i = 0; i < n; ++i)
        {
            for (j = 0; j < n2; ++j)
            {

                // añadir elemento
                sum = sum + list.get(i).get(j);

            }

            subList.add(i, sum);

            // Resetear contador
            sum = 0;
        }
        // ArrayList<Integer> subList = list;
        return subList;
    }
}

```

Bibliografía

- Apache Hive. (2011). Apache Hive TM. Retrieved August 21, 2019, from <https://hive.apache.org/>
- Arigon, A.-M., Tchounikine, A., & Miquel, M. (2006). Handling multiple points of view in a multimedia data warehouse. *ACM Transactions on Multimedia Computing, Communications, and Applications*.
<https://doi.org/10.1145/1152149.1152152>
- Bock, B. C., Páez, V. P., Rand, A. S., & Burghardt, G. M. (2016). LIFE TABLE AND STOCHASTIC MATRIX PROJECTION ANALYSIS FOR A POPULATION OF GREEN IGUANAS (IGUANA IGUANA): IMPLICATIONS FOR CONSERVATION AND CONTROL. In *Herpetological Conservation and Biology* (Vol. 11).
- Cai, Z., Vagena, Z., Perez, L., Arumugam, S., Haas, P. J., & Jermaine, C. (2013). *Simulation of Database-Valued Markov Chains Using SimSQL*.
- Campora, S., Fernandes De Macedo, J. A., & Spinsanti, L. (2011). *St-Toolkit: A Framework for Trajectory Data Warehousing*.
- Chen, P. P.-S. (1976). The entity-relationship model---toward a unified view of data. *ACM Transactions on Database Systems*, 1(1), 9–36.
<https://doi.org/10.1145/320434.320440>
- Ciach, G. J., & Krajewski, W. F. (2002). Radar–Rain Gauge Comparisons under Observational Uncertainties. *Journal of Applied Meteorology*, 38(10), 1519–1525. [https://doi.org/10.1175/1520-0450\(1999\)038<1519:rrgcuo>2.0.co;2](https://doi.org/10.1175/1520-0450(1999)038<1519:rrgcuo>2.0.co;2)
- Gamboa, J. M., & Rodríguez, M. B. (2003). *Álgebra matricial*. Anaya.
- Goicoechea López-Vailo, I., Induráin, E., Azcona Ciriza, E., & Archel, P. (1991). Contabilidad matricial, contabilidad por grafos, matemáticas e informática (hacia un enfoque interdisciplinar).

- Gosain, A., & Singh, J. (2014). Conceptual multidimensional modeling for data warehouses: A survey. *Advances in Intelligent Systems and Computing*. https://doi.org/10.1007/978-3-319-11933-5_33
- Hearon, J. Z. (1976). Properties of the Leslie population matrix. *Bulletin of Mathematical Biology*, 38(2), 199–203. <https://doi.org/10.1007/BF02471756>
- Inmon, W. H. (2002). Building the datawarehouse 3rd Edition. In *Career: Data and Analytics*.
- Jain, N., & Camacho, J. (2017). Enhanced Aggregation, Cube, Grouping and Rollup - Apache Hive - Apache Software Foundation. Retrieved August 21, 2019, from <https://cwiki.apache.org/confluence/display/Hive/Enhanced+Aggregation%2C+Cube%2C+Grouping+and+Rollup>
- Jenhani, F., & Akaichi, J. (2014). *Semantic View on Trajectory Data for Ambulance Services Enhancement: Modeling, Storage, and Analysis Issues*.
- Jensen, C. S., Kligys, A., Pedersen, T. B., & Timko, I. (2002). *Multidimensional Data Modeling for Location-Based Services*.
- Jones, P. G., & Thornton, P. K. (1997). Spatial and temporal variability of rainfall related to a third-order Markov model. *Agricultural and Forest Meteorology*, 86(1–2), 127–138. [https://doi.org/10.1016/S0168-1923\(96\)02399-4](https://doi.org/10.1016/S0168-1923(96)02399-4)
- Kalman, D. (2002). *A Singularly Valuable Decomposition: The SVD of a Matrix*.
- Kimball, R., & Ross, M. (2013). *The data warehouse toolkit : the definitive guide to dimensional modeling*. Wiley.
- Lucía Urueña, O. B., & San Mateo, F. (n.d.). *CONTABILIDAD BÁSICA René Descartes COLECCIÓN DIDÁCTICA CIENCIAS ECONÓMICAS Y ADMINISTRATIVAS Fundación para la Educación Superior San Mateo Bogotá D.C. Colombia Sello Editorial*.
- Luo, S., Gao, Z. J., Gubanov, M., Perez, L. L., & Jermaine, C. (2018). Scalable linear algebra on a relational database system. *SIGMOD Record*, 47(1), 24–31. <https://doi.org/10.1109/ICDE.2017.108>
- Malinowski, E., & Zimanyi, E. (2005). *Representing spatiality in a conceptual*

- multidimensional model*. <https://doi.org/10.1145/1032222.1032226>
- Malinowski, E., & Zimanyi, E. (2007). *Implementing spatial datawarehouse hierarchies in object-relational DBMSs. RDF2OLAP View project Modelling Data Warehouses with Multiversion and Temporal Functionality View project IMPLEMENTING SPATIAL DATA WAREHOUSE HIERARCHIES IN OBJECT-RELATIONAL DBMSs*. Retrieved from <https://www.researchgate.net/publication/220709384>
- Malinowski, E., & Zimányi, E. (2005). *Spatial Hierarchies and Topological Relationships in the Spatial MultiDimER Model*. https://doi.org/10.1007/11511854_2
- Malinowski, E., & Zimányi, E. (2008). Designing Spatial and Temporal Data Warehouses. In *Advanced Data Warehouse Design* (pp. 315–352). https://doi.org/10.1007/978-3-540-74405-4_7
- Moreno, F., Alberto, J., Arias, E., & Losada, B. M. (2010). *A CONCEPTUAL SPATIO-TEMPORAL MULTIDIMENSIONAL MODEL* (Vol. 9).
- Moreno, F., Echeverri, J., & Arango, F. (2010). Reclassification queries in a geographical data warehouse. *Revista Técnica de La Facultad de Ingeniería Universidad Del Zulia*, 33, 263–271.
- Orlando, D., Frade, A., Nelson, J., & Castillo, P. (2007). Estado actual de las tecnologías de bodega de datos y Estado actual de las tecnologías de bodega de datos y OLAP aplicadas a bases de datos espaciales OLAP aplicadas a bases de datos espaciales Current data warehousing and OLAP technologies' status applied to spatial databases. In *ABRIL DE* (Vol. 27).
- Orlando, S., Orsini, R., Raffaeta, A., Roncato, A., & Silvestri, C. (2012). Trajectory Data Warehouses: Design and Implementation Issues. *Journal of Computing Science and Engineering*. <https://doi.org/10.5626/jcse.2007.1.2.211>
- Oueslati, W., & Akaichi, J. (2014). *Trajectory data warehouse modeling based on a Trajectory UML profile: Medical example*.
- Paradigm4. (2019). Paradigm4: Creators of SciDB a computational DBMS. Retrieved August 22, 2019, from <https://www.paradigm4.com/>

- Parent, C., Spaccapietra, S., & Zimányi, E. (2004). *Spatio-temporal conceptual models*. <https://doi.org/10.1145/320134.320142>
- Poca Freddy, C. (2017). *Contabilidad Matricial* (Vol. 2).
- Prabhune, O., Sabale, P., Sonawane, D. N., & Prabhune, C. L. (2017). Image processing and matrices. *2017 International Conference on Data Management, Analytics and Innovation, ICDMAI 2017*, 166–171. <https://doi.org/10.1109/ICDMAI.2017.8073504>
- Rajadell, M., Trullàs, O., & Simo, P. (2014). Contabilidad para todos: Introducción al registro contable. In *Contabilidad para todos: Introducción al registro contable*. <https://doi.org/10.3926/oss.14>
- Reinosa, E. J. (2012). *Bases de datos*. Alfaomega.
- Silvers, F. (2008). *Building and maintaining a data warehouse*. CRC Press.
- Spaccapietra, S., Parent, C., Damiani, M. L., de Macedo, J. A., Porto, F., & Vangenot, C. (2008). A conceptual view on trajectories. *Data and Knowledge Engineering*. <https://doi.org/10.1016/j.datak.2007.10.008>
- Steinbach, C., & Coppage, K. (2019). LanguageManual Types - Apache Hive - Apache Software Foundation. Retrieved August 21, 2019, from <https://cwiki.apache.org/confluence/display/Hive/LanguageManual+Types>
- TERADATA. (n.d.). Array (Teradata Database) — Teradata Warehouse Miner User Guide - Volume 2ADS Generation. Retrieved August 21, 2019, from <https://docs.teradata.com/reader/mzKiGa9RuIQFqqGK3GEgrQ/AlFymvwOyu5Bm2dRfMQH9A>
- TERADATA. (2018). ARRAY/VARRAY Data Type — Teradata Database SQL Data Types and Literals. Retrieved August 21, 2019, from https://docs.teradata.com/reader/S0Fw2AVH8ff3MDA0wDOHIQ/u~IAp_JFEOcP~KfcOCOGsA
- The Apache Software Foundation. (2015). Apache SystemML - Declarative Large-Scale Machine Learning. Retrieved August 22, 2019, from <https://systemml.apache.org/>
- Tryfona, N., Busborg, F., & Borch Christiansen, J. G. (2004, February 3). *starER*.

- 3–8. <https://doi.org/10.1145/319757.319776>
- Wagner, R., de Macedo, J. A. F., Raffaetà, A., Renso, C., Roncato, A., & Trasarti, R. (2014). *Mob-Warehouse: A Semantic Approach for Mobility Analysis with a Trajectory Data Warehouse*. https://doi.org/10.1007/978-3-319-14139-8_15
- Yan, Z., Parent, C., Spaccapietra, S., & Chakraborty, D. (2010). A hybrid model and computing platform for spatio-semantic trajectories. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-642-13486-9_5
- Zimányi, E. (2012). Spatio-temporal datawarehouses and mobility data: Current status and research issues. *Proceedings - 2012 19th International Symposium on Temporal Representation and Reasoning, TIME 2012*, 6–9. <https://doi.org/10.1109/TIME.2012.29>