

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/352713986>

Diseño de bases de datos para uso de big data en tiempo real

Chapter · January 2018

CITATIONS

0

READS

35

2 authors:



Diego Ibarra-Corona

Autonomous University of Queretaro

9 PUBLICATIONS 3 CITATIONS

[SEE PROFILE](#)



Jose Alejandro Vargas-Diaz

Autonomous University of Queretaro

13 PUBLICATIONS 1 CITATION

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Diseño de una Red Organizacional de Conocimiento [View project](#)

DISEÑO DE BASES DE DATOS PARA USO DE BIG DATA EN TIEMPO REAL

Diego Octavio Ibarra Corona

Eduardo Aguirre Caracheo

Jesús Armando Rincones

José Alejandro Vargas Díaz

RESUMEN

Las bases de datos (BD) permiten almacenar grandes volúmenes de información ordenadamente, y en particular benefician a las organizaciones en sus procesos. Las bases de datos constan de dos componentes principales: un modelo y un motor. El modelo indica la manera en que se organizan los datos, y tiene gran influencia sobre la cantidad de tiempo que toma buscar los datos almacenados. El motor es el componente responsable de realizar operaciones de búsqueda, agregando, modificando o eliminando los datos. Una base de datos generalmente se alimenta de datos de una fuente constante y uniforme, aunque también es posible obtener datos de diversas fuentes. En tales casos, el volumen de datos que se obtiene puede llegar a ser masivo. Cuando se trabaja con un volumen de datos amplio, las operaciones sobre los datos pueden llegar a perder velocidad, por lo que es recomendable seleccionar un modelo y un motor de base de datos adecuado para solventar dicho problema. En este trabajo de investigación se analizan diferentes modelos y motores con el fin de identificar los más adecuados para trabajar con volúmenes masivos de datos en el menor tiempo posible.

1. INTRODUCCIÓN

El tratamiento de la información es el área de la informática que se dedica al estudio y manejo de datos provenientes de diferentes fuentes. El uso y manejo de la información es indispensable para que las empresas puedan comprender mejor el funcionamiento de sus procesos actuales, lo cual les permite encontrar áreas de oportunidad para mejorar procesos, metodologías, o incluso sus servicios y productos. Mientras mayor sea la cantidad de información que almacene y procese una empresa, deberán ser mejores las herramientas con las que cuente para aprovecharla. Los datos pueden extraerse de fuentes muy diversas y pueden abarcar muchas áreas, por lo cual el volumen puede llegar a ser tan amplio que su almacenamiento y procesamiento resulte ser complejo.

Por lo tanto, se debe definir una estructura adecuada para almacenar dichos datos, así como un mecanismo que permita su manipulación de forma ágil para permitir a la empresa agruparlos, analizarlos y aprovecharlos para la mejora de sus procesos internos. Este capítulo abarca los conceptos necesarios para identificar los diferentes modelos que existen para organizar información, así como los distintos motores que existen para procesarlos, en búsqueda de la combinación más adecuada de ambos, que contribuya al acceso ágil de grandes volúmenes de datos almacenados que permitan a la empresa tomar decisiones operativas y administrativas.

2. MARCO TEÓRICO

2.1. BASE DE DATOS

Una Base de datos (BD) es un repositorio electrónico de información. En las BD, los datos se almacenan en estructuras denominadas tablas, las cuales organizan los datos en entidades. Una entidad es un objeto de la vida real, y en las BD existe una tabla por cada entidad. Generalmente, dichas entidades no existen aisladas: pueden existir relaciones entre ellas, las cuales también se expresan en la BD. Para Gyorödi y Gyorödi (2015) el principal propósito de las BD es almacenar la información de las entidades que una empresa requiere para funcionar. Las BD almacenan los datos de los procesos operativos de la empresa, permitiéndole a los altos mandos tomar decisiones administrativas y operativas de acuerdo a estos datos.

Resultan útiles cuando múltiples usuarios requieren de acceso a los mismos datos de manera concurrente. Las BD almacenan datos en archivos en el

disco duro de una computadora con la finalidad de proveer a los usuarios de una manera sencilla de obtener información sobre esos datos (Date, 2001). La gran mayoría de los usuarios se comunica con las BD a través de un programa intermediario, que ejecuta las consultas (o solicitudes de información) que realizan los usuarios, transmiten dicha consulta a la BD, y muestra al usuario la respuesta que arroja la bd en base a su consulta (Bing y Hevner, 1987). Dichas consultas son escritas en un lenguaje conocido como SQL (Structured Query Language), que permite la interacción de los usuarios con la BD.

Las BD generalmente se alojan en un tipo de computadora especial, denominada servidor, que tiene la capacidad de atender las peticiones de múltiples usuarios con gran agilidad, además de contar con un amplio espacio de almacenamiento capaz de soportar el volumen de información que alojará la BD (Ailamaki, 2011). Si bien es común hacer la instalación del servidor de BD dentro del edificio de la empresa que la requiere, también se puede alojar en una ubicación física ajena a la empresa, siendo la BD accedida remotamente a través de una red (Sharma y Singh, 2012).

Pérez (2015) menciona que las BD tienen dos funciones principales: almacenar todos los datos que genera una empresa como resultado de sus operaciones, y proveer a los empleados de dicha empresa de un mecanismo para acceder a estos datos, a fin de obtener la información requerida para continuar con sus operaciones. Para Date (2001) generalmente cuentan con un usuario especial, conocido como el Administrador de Base de Datos, o DBA, que se encarga de gestionar el acceso a los datos por parte de los demás usuarios, de implementar las medidas de seguridad necesarias, y asegurarse de que el rendimiento y funcionamiento de la BD sea óptimo en todo momento. Al ser un repositorio centralizado, los empleados de la empresa también pueden acceder a los datos como parte de sus procesos, sea para consultarlos, sea para agregarlos. Por lo tanto, es posible tener más de un usuario accediendo a la BD, y en ocasiones el número de usuarios conectados a la BD de forma concurrente puede llegar a ser muy alto. Esto puede provocar que el acceso para consulta o agregado de datos sea ineficiente, al tener varios usuarios intentando realizar una conexión a un punto común. Por lo tanto, previo a la creación de la BD, es preciso definir un modelo que permita un flujo eficiente de información entre la BD y los usuarios.

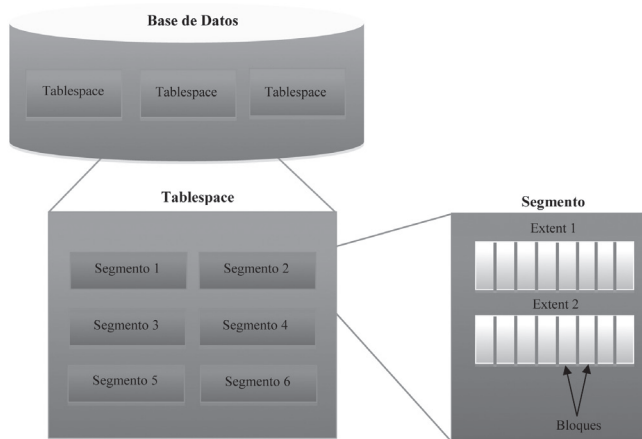
La interacción y manejo de datos con una BD generalmente se hace mediante un Sistema Manejador de Base de Datos (SMBD). Éste es un software encargado de permitir la comunicación de los usuarios finales con la BD, así como la creación, consulta de información y administración de la BD. El SMBD generalmente se instala en el servidor que aloja la BD. Existen distintos tipos

de BD, cada una de las cuales requiere su propio SMBD compatible (Ullman y Widom, 1997).

En el SMD de Oracle, las BD utilizan diversas estructuras de almacenamiento donde se conservan los datos. Dichas estructuras pueden ser tanto físicas como lógicas. Un bloque es la unidad lógica de almacenamiento más pequeña, cuyo tamaño se mide en bytes. El tamaño del bloque generalmente es un múltiplo del tamaño de bloque del sistema operativo que aloja la BD, lo cual hace más eficientes las operaciones de entrada y salida del disco duro. Una colección de bloques crea un *extent*, lo cual representa una porción de espacio libre para almacenar datos.

Un conjunto de *extents* crea un segmento, lo cual representa un objeto que la BD trata como una unidad única, ya sea una tabla o un índice. Los segmentos son la unidad de almacenamiento más pequeña con la que tratan los usuarios de las BD. Uno o varios segmentos pueden almacenarse dentro de un *tablespace*, el cual representa el área de trabajo de un determinado usuario que se conecta a la BD (Bryla y Loney, 2008). Generalmente, cada usuario accede sólo a los objetos ubicados dentro de su propio *tablespace*, aunque es posible acceder a los objetos ubicados en el *tablespace* de otro usuario, de tener los permisos de acceso correspondientes.

FIGURA 1. ESTRUCTURAS DE ALMACENAMIENTO LÓGICAS DE LAS BD EN ORACLE¹



¹ Fuente: elaboración propia, con base en Bryla y Loney, 2008.

Las estructuras lógicas de las BD son las tablas, la unidad básica de almacenamiento. Son conjuntos de filas y columnas que ofrecen características robustas como confiabilidad, integridad y escalabilidad ausentes en otras estructuras similares para almacenamiento de datos, como las hojas de cálculo o los archivos de texto. Las tablas permiten organizar información de diversos tipos, tales como cadenas de caracteres, números, fechas, etc. En las tablas, los datos generalmente se almacenan sin orden particular, siendo deber del motor de la BD ordenar los datos con base en algún criterio proporcionado por el usuario. Las tablas pueden almacenar información tanto de procesos internos de la empresa, como de fuentes externas, tales como las hojas de cálculo o archivos de texto antes mencionados, e incluso otras tablas.

Además de las estructuras lógicas, las bases de datos se conforman por una determinada estructura física que rige, tanto el almacenamiento de la información como la comunicación con los usuarios con base en consultas en lenguaje SQL. En el caso de las bases de datos utilizadas con el SMBD llamado MySQL, la estructura física mostrada en la Figura 2, se divide en: capa de aplicaciones e interfaces, capa de procesador de consulta, capa del manejador de transacción y recuperación, y, capa del manejador de almacenado.

CAPA DE APLICACIONES E INTERFACES

Es la capa donde tanto las aplicaciones como los usuarios interactúan con el SMBD de MySQL. Los diferentes usuarios que se consideran en esta capa son los administradores, los clientes y los usuarios de consultas.

CAPA DE PROCESADOR DE CONSULTA

Se conforma de diversos componentes que permiten la interacción con el sistema, cuando el usuario realiza una consulta SQL. Dichos componentes se describen a continuación:

- a) Precompilador DML: extrae las instrucciones SQL incrustadas en las API del cliente cada vez que se recibe una petición.
- b) Recopilador DDL: compila instrucciones SQL y las prepara su ejecución.
- c) Analizador de consulta: analiza la consulta SQL y crea una estructura de análisis para facilitar su procesamiento.

- d) Preprocesador de consulta: comprueba la sintaxis de SQL para comprobar si su semántica es válida.
- e) Manejador de seguridad: una vez que la consulta se determina válida, se determina si el usuario tiene privilegios de acceso a la base de datos.
- f) Optimizador de consulta: analiza la consulta procesada para verificar si puede optimizarse en cualquier forma que permita agilizar su procesamiento.
- g) Motor de ejecución: ejecuta la consulta, al tener acceso a la capa física de la BD.

CAPA DEL MANEJADOR DE TRANSACCIÓN Y RECUPERACIÓN

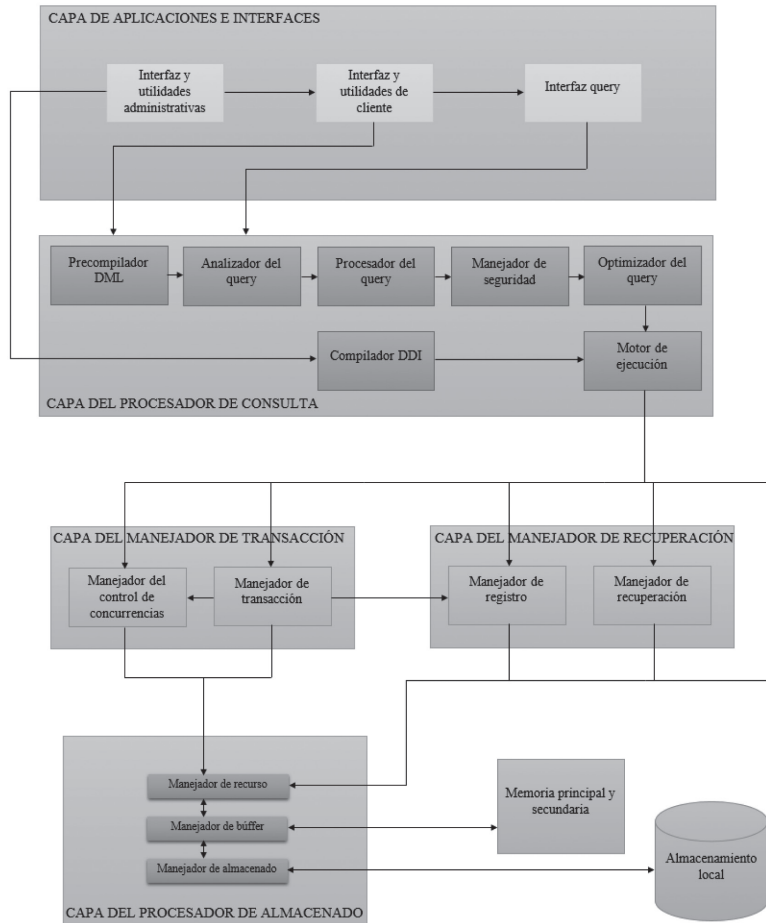
- a) Manejador de transacción: se asegura de que el conjunto de comandos SQL se ejecute de forma atómica.
- b) Manejador del control de concurrencias: se asegura de que los datos de la BD sólo sean modificados por una consulta SQL a la vez, de forma que se evite el bloqueo de datos por accesos simultáneos.
- c) Manejador del registro: almacena las instrucciones SQL en un área de memoria temporal para que pueda recuperarse dicha información en caso de que exista algún fallo en la BD.
- d) Manejador de recuperación: restaura la BD a un estado anterior, ejecutando cada operación almacenada en el registro.

CAPA DEL MANEJADOR DE ALMACENADO

Esta capa es responsable de efectuar almacenamiento secundario por medio de almacenadores intermediarios, los cuales residen en la memoria principal.

- a) Manejador de almacenamiento: se encarga de recuperar los datos del disco físico y los propaga al manejador de búffer.
- b) Manejador del búffer: determina cuánta memoria asignar para la manipulación de datos.
- c) Manejador de recurso: recibe las referencias a los datos de la memoria del manejador de almacenamiento y los arroja a las capas superiores.

FIGURA 2. ESTRUCTURA FÍSICA DE MYSQL²



2.2. MODELO DE BASE DE DATOS

Un modelo de base de datos es el diseño en el que se plantea organizar la información en una BD. Dicho modelo generalmente es una representación visual de la manera en que se organizarán los datos, y se crea en una etapa de análisis

² Fuente: elaboración propia con base en:
<https://tallerbd.wikispaces.com/-+Estructura+de+MySQL>

de requerimientos. El modelo de BD permite visualizar todas las entidades y las relaciones entre estas para observar que se contemplen todos los requisitos de datos de la empresa. Si existe algún dato o entidad que no se esté considerando en el modelo, es durante esta etapa de análisis cuando es más sencillo y menos costoso modificarlo para agregar cualquier variable faltante.

El modelo de una BD es el diseño que indica la forma en que se van a almacenar los datos dentro de la BD (Taniar, 2012). Dicho diseño debe satisfacer las siguientes condiciones: describir qué datos se contienen en la base de datos; describir las relaciones (si las hay) que existen entre los diferentes datos, y describir las reglas que deben cumplir sobre los datos (Bing y Hevner, 1987). El modelo de la BD es una representación formal de la estructura que almacenará los datos, y permite al desarrollador de software verificar si se cumplen todos los requerimientos del usuario (Hansen y Hansen, 2010).

El modelo de una BD representa la estructura lógica que indica la forma en que los datos serán almacenados, manejados y procesados. Por lo tanto, dicho diseño es considerado como los cimientos de la BD (Date, 2001). En el modelo se incluyen las entidades involucradas, las relaciones entre entidades, y los objetos necesarios para satisfacer las reglas de negocio de la empresa que requiere la BD (Watt y Eng, 2014). Existen diferentes tipos de modelos de base de datos, cada uno de ellos con ventajas e inconvenientes que los hacen adecuados para contextos y ambientes distintos (Chun, Kwee, *et al.*, 2009).

Uno de los modelos más comunes es el relacional. Dicho modelo plantea que todas las entidades deben estar asociadas entre sí, y funciona de manera eficiente cuando el conjunto de datos existente es limitado. Otro modelo poco más reciente es el modelo no relacional, que dicta que los datos no se representan en tablas, sino en estructuras flexibles, y no necesariamente debe haber una relación entre ellas. En este modelo, la información simplemente se busca con base en llaves de identificación, y es más adecuado cuando se considera que la cantidad de datos que se pretende almacenar crezca exponencialmente. A diferencia del modelo relacional, no existe complicación alguna al agregar o remover variables del conjunto de datos, ya que la estructura de la BD no se encuentra definida (Gyorödi y Gyorödi, 2015).

Otro modelo utilizado para las BD es el de normalización. Su propósito es crear un esquema relacional que evite redundancias de datos, sin perder la facilidad de búsqueda de datos. De acuerdo con Date (2001), una forma de evitar la presencia de datos duplicados en una BD es colocando “cada hecho en su lugar”, es decir, definiendo cada tabla sólo con los campos que estrictamente necesita, y colocar el resto de los campos en una entidad separada. En la Figura 3 se muestra el diseño de la BD de una tienda departamental de ropa sin

normalizar, lo cual significa que se almacena información de diversos hechos dentro de una misma tabla. Dicha tabla almacena la información del artículo de ropa que compra cada uno de los clientes. El diseño sin normalizar genera gran redundancia de datos.

FIGURA 3. DISEÑO SIN NORMALIZAR³

ID_USUARIO	NOMBRE_USUARIO	ID_PRODUCTO	PRODUCTO
1	Carlos López	5	Suéter
1	Carlos López	23	Pantalón
1	Carlos López	83	Pantalón
1	Carlos López	905	Camisa
2	Karla Rodríguez	47	Bufanda

La Figura 4 ilustra la misma BD de una tienda de ropa, esta vez con un diseño normalizado. Dicho diseño dicta que cada una de las entidades debe residir dentro de su propia tabla, lo cual permite tener un mejor orden de los datos, así como un nivel de redundancia mucho menor a comparación de un modelo sin normalizar.

FIGURA 4. DISEÑO NORMALIZADO⁴

ID_USUARIO	USUARIO
1	Carlos López
2	Karla Rodríguez

ID_PRODUCTO	PRODUCTO
5	Suéter
23	Pantalón
47	Bufanda
83	Pantalón
905	Camisa

³ Fuente: elaboración propia.

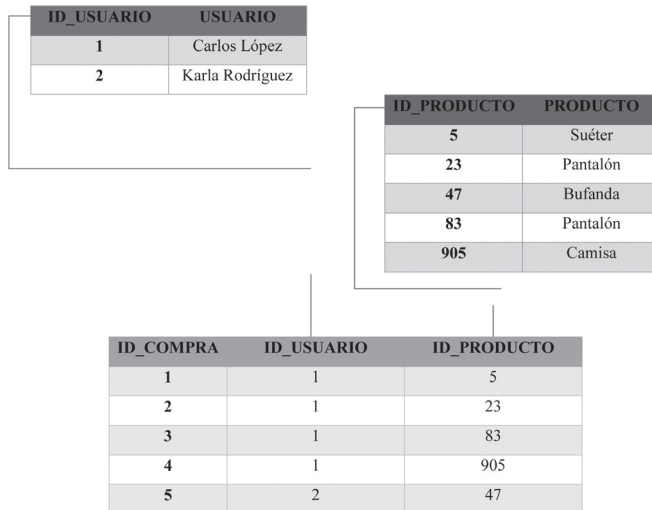
⁴ Fuente: elaboración propia.

ID_COMPRA	ID_USUARIO	ID_PRODUCTO
1	1	5
2	1	23
3	1	83
4	1	905
5	2	47

De acuerdo con Kent (1982), la normalización es una serie de reglas que rigen el modelado de los datos en la BD para evitar errores e inconsistencias en su manejo. Dichos beneficios se proveen a cambio de un tiempo de recuperación de datos ligeramente mayor. Puede comprobarse que el proceso de normalización se ha realizado correctamente, si al devolver los datos a un modelo sin normalizar no existe pérdida de información.

El uso de un diseño normalizado implica utilizar un diseño relacional. En la figura 5 se ilustra la manera en que diferentes tablas se relacionan entre sí mediante el uso de columnas en común, lo cual indica conexiones entre ellas. Dichas relaciones se establecen mediante las llaves de identificación de cada una de las tablas.

FIGURA 5. DISEÑO RELACIONAL⁵



⁵ Fuente: elaboración propia.

MOTORES DE BASE DE DATOS

El motor de una BD es el software encargado de procesar los datos almacenados. Dicho motor es el responsable de hacer la recuperación, inserción, modificación y eliminación de los datos (Date, 2001). Se compone de dos partes: el primer componente es el motor de almacenamiento, que se encarga tanto de escribir como de recuperar los datos del disco duro; el segundo componente es el procesador de consultas, quien recibe, interpreta y ejecuta instrucciones en lenguaje SQL, con el cual se escriben las consultas para la base de datos (Chun, Kwee, *et. al.*, 2009). A pesar de que cada diferente motor de BD consta de los componentes antes mencionados, cada motor funciona de manera distinta, lo cual permite que cada uno de ellos tenga desempeño diferente, dependiendo del ambiente donde se ubique (Mukherjee, 2013).

INFORMACIÓN EN TIEMPO REAL

Si bien es cierto que las empresas dependen de los datos almacenados en su BD para comprender el funcionamiento actual de sus procesos, ver las áreas de oportunidad para mejorar, y tomar decisiones en la alta gerencia. También es cierto que la información se mantiene resguardada en la BD hasta que algún usuario efectúa una consulta sobre ella (Huiming y Zao, 2010). Esto repercute en que la información que se extraiga de la BD no siempre se encuentre actualizada al minuto de la consulta, lo que en algunos negocios puede generar resultados que no reflejen la realidad del momento.

La capacidad de acceder a la información almacenada en la BD de la forma más ágil posible implica obtener resultados certeros y actuales, lo que en algunos contextos es de vital importancia para el buen funcionamiento de la empresa (Nimalasena, 2014). Pueden existir casos en los que los datos que almacena la BD se actualicen con tal frecuencia que, al momento de realizar el análisis de una consulta de información, ésta se haya vuelto obsoleta. Es en estos casos donde la habilidad de efectuar el análisis de datos en tiempo real proveerá a los usuarios, generalmente directivos, información al minuto, de forma que se tenga un panorama claro y actual sobre el comportamiento de los datos, a fin de tomar las decisiones estratégicas y administrativas más apropiadas para la empresa, con base en este análisis en tiempo real (Huiming y Zao, 2010).

BIG DATA

El término Big Data se refiere a un conjunto de información con un volumen mucho más grande de lo habitual. Por lo tanto, se requiere de un sistema más robusto para soportar la mayor cantidad de datos, compuesto tanto de software especializado, como de herramientas que permitan la captura, almacenamiento, recuperación y análisis de los datos (More y Rajendra, 2015). Las fuentes de Big Data pueden llegar a ser muy variadas, y la velocidad con la que los datos se almacenan y recuperan debe ser equivalente a la de una BD común para que sea de utilidad.

Una de las principales características del Big Data es su capacidad de ser obtenida de fuentes de datos no convencionales (Pérez, 2015). Mientras una empresa puede generar sus propios datos con base en sus procesos operativos para posteriormente alojarlos en su BD, el Big Data puede provenir de fuentes de información tales como redes sociales, páginas de Internet, sensores de maquinaria o registros de auditoría. Es por ello que, al tener mayores fuentes de datos, el volumen de información que se genera es mayor del que puede encontrarse en una BD tradicional (Shaw, 2015).

El Big Data ha ido creciendo en popularidad dentro de las organizaciones por el valor que provee: una mayor cantidad de datos permite a las empresas obtener más información, la cual puede ser aprovechada para mejorar sus procesos internos, extender sus relaciones con clientes y proveedores, y proveer una mejor visión de las tendencias que se presentan en el entorno para adecuar los recursos y procesos de manera correspondiente (Pérez, 2015).

3. METODOLOGÍA

La metodología siguió un enfoque cuantitativo para el cual fueron necesarios los siguientes recursos: un servidor de BD con amplio espacio en disco duro para almacenar el volumen de datos necesario para considerarse Big Data; una computadora que actúe como cliente, realizando peticiones al servidor de BD, y capital humano para utilizar los recursos antes mencionados y efectuar la medición de variables. Las variables por medir fueron los diferentes modelos de BD, los distintos motores de BD, la cantidad de información por almacenar en el servidor y los tiempos de ejecución de consultas.

Se realizaron diversas combinaciones de modelos y motores de BD, todas incluyendo la misma información en volumen de Big Data. Posteriormente, se compararon los tiempos de recuperación de datos para cada combinación, con

el fin de identificar la que proporcione los tiempos más bajos. La forma de medir dichos tiempos de recuperación de datos fue mediante consultas en lenguaje SQL y cronometrando el tiempo que toman para devolver la información solicitada. Las consultas se hicieron de la manera más similar posible entre las diferentes combinaciones de modelos y motores para asegurar un resultado equivalente.

Se consideraron tres modelos de BD: relacional, no relacional y normalizado; del mismo modo se considerarán tres motores de BD: Oracle, MySQL y PostgreSQL. Las BD se llenaron con un volumen masivo de datos climatológicos, sobre los cuales se efectuarán consultas para medir los tiempos de recuperación. Posteriormente, se compararán dichos tiempos entre los diferentes modelos y motores en busca del más bajo. El resultado que se busca obtener es el modelo con el correspondiente motor que permita recuperar los datos de la manera más ágil. Se realizó una consulta en lenguaje SQL, aplicando un filtro, dependiendo de la cantidad de datos por recuperar. Cada una de las consultas de los diferentes modelos se realizó en un arranque en frío del servidor para asegurar la consistencia de los resultados. Las consultas requeridas para cronometrar el tiempo fueron tan similares como sea posible entre motores, considerando las diferencias de sintaxis propias de dichos diferentes motores.

4. RESULTADOS

La Tabla 1 representa los tiempos de recuperación de datos con cada diferente motor de BD, utilizando un modelo relacional.

TABLA 1. TIEMPOS DE RECUPERACIÓN DE DATOS
CON UN MODELO RELACIONAL⁶

VOLUMEN DE DATOS	ORACLE	mysql	POSTRESQL
50,000 registros	36.623 seg	6.170 seg	13.341 seg
75,000 registros	40.844 seg	6.986 seg	18.563 seg
100,000 registros	52.328 seg	7.324 seg	24.228 seg
150,000 registros	77.976 seg	9.214 seg	35.472 seg
200,000 registros	150.234 seg	15.455 seg	47.234 seg

⁶ Fuente: elaboración propia.

La Tabla 2 representa los tiempos de recuperación de datos con cada diferente motor de BD, utilizando un modelo no relacional.

TABLA 2. TIEMPOS DE RECUPERACIÓN DE DATOS
CON UN MODELO NO RELACIONAL⁷

VOLUMEN DE DATOS	ORACLE	mysql	POSTRESQL
50,000 registros	30.549 seg	3.749 seg	11.435 seg
75,000 registros	35.014 seg	3.780 seg	15.335 seg
100,000 registros	45.315 seg	3.874 seg	21.825 seg
150,000 registros	68.688 seg	5.233 seg	32.713 seg
200,000 registros	144.911 seg	10.514 seg	43.040 seg

La Tabla 3 representa los tiempos de recuperación de datos con cada diferente motor de BD, utilizando un modelo normalizado.

TABLA 3. TIEMPOS DE RECUPERACIÓN DE DATOS
CON UN MODELO NORMALIZADO⁸

VOLUMEN DE DATOS	ORACLE	mysql	POSTRESQL
50,000 registros	48.451 seg	9.542 seg	19.235 seg
75,000 registros	53.643 seg	10.762 seg	29.640 seg
100,000 registros	63.123 seg	11.456 seg	35.008 seg
150,000 registros	87.987 seg	12.342 seg	47.025 seg
200,000 registros	191.634 seg	20.540 seg	52.203 seg

5. CONCLUSIONES

Los diferentes modelos de BD dictan distintas formas de estructurar un mismo conjunto de datos. Con base en el motor que se elija para realizar las opera-

⁷ Fuente: elaboración propia.

⁸ Fuente: elaboración propia.

ciones de lectura, inserción, modificación y eliminación variará la cantidad de tiempo que demoren dichas tareas. Con base en esta investigación, se obtiene que el modelo más adecuado para manejar grandes volúmenes de datos es el no relacional. Al no existir relaciones entre las diversas tablas, el motor de BD no debe realizar las correspondientes uniones y validaciones entre objetos, lo cual tuvo como resultado un tiempo de obtención de datos menor. Trabajando con modelos no relacionales, el motor de BD que obtuvo los tiempos de búsqueda más bajos fue MySQL. Por lo tanto, dicho motor almacenando datos en un modelo no relacional resulta el más adecuado para trabajar con volúmenes masivos de datos en el menor tiempo posible.

Las BD son herramientas indispensables en la labor de cualquier organización, entidad o sistema informático, por lo tanto, conocer a profundidad su funcionamiento permite aprovechar mejor sus características. En este trabajo de investigación se compararon tres modelos diferentes, a manera de organizar un mismo conjunto de datos; de igual modo, se comparó el desempeño de tres diferentes motores de BD, utilizando cada uno de dichos modelos con los modelos mencionados. Los diversos modelos y motores tienen aplicaciones diferentes, debiéndose seleccionar el que sea más adecuado, según el requerimiento: sea tener una mejor organización de los datos o un menor tiempo de recuperación de ellos.

REFERENCIAS

- AILAMAKI, A., *Embarrassingly Scalable Database Systems*, IEEE, Estados Unidos, 2011.
- BING, S. y Hevner, A., *Analysis of Database System Architectures Using. Benchmark*, IEEE, Estados Unidos, 1987.
- CHUN, O., Kwee, A. y Tsai, F., *Database Optimization for Novelty Detection*, IEEE, Singapur, 2009.
- D NATE, C., *Introducción a los sistemas de bases de datos*, Sage, Estados Unidos, 2001.
- GYORÖDI, C. y Gyorödi R., *A Comparative Study: MongoDB vs. MySQL*, IEEE, Estados Unidos, 2015.
- HANSEN, G. y Hansen, J., *Diseño y Administración de Bases de Datos*, Prentice Hall, Estados Unidos, 2010.
- HUIMING, L. y Zhao, Z., *The Researching and Application of Historical Data Processing In Real-Time Database System*, IEEE, Estados Unidos, 2010.

- Kent, W., "A Simple Guide to Five Normal Forms in Relational Database Theory", *Communications of the ACM*, IEEE, Estados Unidos, No. 26, 1982, pp. 120-125.
- MORE, S. y Rajendra, H., *Artificially Intelligent Optimized Database (AIOD)*, IEEE, Estados Unidos, 2015.
- MUKHERJEE, R., *Interfacing data destinations and visualizations: A history of database literacy*, Sage, Estados Unidos, 2013.
- NIMALASENA, A., *Performance Tuning of Database Systems Using a Context-aware Approach*, IEEE, Estados Unidos, 2014.
- PÉREZ, M.M., *Big Data: técnicas, herramientas y aplicaciones*, Alfaomega, México, 2015.
- ULLMAN, J. y Widom, J., *A First Course in Database Systems*, Prentice Hall, Estados Unidos, 1997.
- SHARMA, M. y Singh, G., *Analysis of Joins and Semi-joins in Centralized and Distributed Database Queries*, International Conference on Computing Sciences, Estados Unidos, 2012.
- SHAW, R., *Big Data and reality*, Sage, Estados Unidos, 2015.
- TANIAR, D., *High Performance Database Processing*, IEEE, Estados Unidos, 2012.
- WATT, A. y Eng, N., *Database Design*, Open Textbook Project, Estados Unidos, 2014.