

EVALUACIÓN DE LOS SISTEMAS DE ADMINISTRACIÓN DE BASES DE DATOS CON EXTENSIONES ESPACIALES

Elzbieta Malinowski

Resumen

La práctica común para la creación de repositorios de datos espaciales es el uso de archivos *shape*, los cuales no proporcionan características necesarias para el control de la consistencia de los datos, la redundancia o el acceso concurrente, entre otros. Por otra parte, las extensiones espaciales de los sistemas de administración de bases de datos (SABD) facilitan el uso de los mecanismos de controles declarativos o dinámicos (disparadores), aplicados a datos convencionales y espaciales. Sin embargo, SABD no siempre es considerado en el desarrollo de sistemas de información espacial o infraestructuras de datos espaciales debido a que la mayoría de este tipo de sistemas requiere el conocimiento de geo-especialistas y ellos están generalmente más familiarizados con los sistemas de información geográfica que con los SABD. Además, aunque existen varios SABD con extensiones espaciales, estos todavía no han sido evaluados o comparados para facilitar el proceso de selección de uno de ellos de acuerdo con las necesidades de los usuarios y aplicaciones. En este artículo se discute la necesidad de considerar SABD como una parte de los sistemas de información espacial y se evalúan cuatro sistemas: PostgreSQL/PostGIS, MySQL, Oracle y SQL Server. Se utilizan los criterios de evaluación desarrollados con base en los estándares de datos espaciales y también se comparan los requisitos de almacenamiento para los índices espaciales y las relaciones que contienen datos espaciales. Además, se aplica el proceso de comparación a los tiempos de ejecución para diferentes grupos de consultas. La evaluación realizada claramente favorece a PostGIS con respecto a tipos de geometría y métodos usados como también a los indicadores de almacenamiento y desempeño de las consultas. Sin embargo, en la decisión de utilizar este sistema como alternativa de almacenamiento de datos espaciales se pueden considerar otros aspectos mencionados al final del presente artículo.

Palabras claves: Sistemas de Administración de Bases de Datos (SABD), Sistemas de Gestión de Bases de Datos (SGBD), Evaluación de SABD con extensiones espaciales, Bases de datos espaciales, Datos espaciales, Operaciones espaciales.

Abstract

The common practice for creating a spatial data repository is to use shape files that do not provide features for controlling data consistency, redundancy, or concurrent access, among others. On the other hand, current spatial extensions to database management systems (DBMSs) facilitate the use of well-known features, e.g., declarative or dynamic (triggers) constraints, applied to conventional and spatial data. Nevertheless, these kinds of systems are not always taken into consideration when building spatial information systems or spatial data infrastructures given the fact that most of this kind of systems requires knowledge of geo-specialists and they are usually more familiar with Geographic Information Systems than DBMSs, choosing the former as a storage platform for spatial applications. Furthermore, even though there are several spatially-extended DBMSs, to our knowledge they have not yet been evaluated or compared to facilitate the selection process to use them. In this paper, we discuss the need to consider DBMSs with spatial extensions as part of spatial information systems or spatial data infrastructures and we evaluate four systems, i.e., PostgreSQL/PostGIS, MySQL, Oracle, and SQL Server. We use evaluation criteria that we developed based on spatial data standards and also examine storage requirements for spatial indexes and relations with spatial instances. The evaluation clearly favors PostGIS regarding geometry types and methods, as well as storage and performance indicators. Nevertheless, the decision to use this system as an alternative for spatial data storage may consider other aspects mentioned in the final part of this article.

Keywords: Spatial DBMS evaluation, spatial data, spatial operations.

Recibido: 30 de octubre de 2013 • **Aprobado:** 13 de febrero de 2014



1. INTRODUCCIÓN

El software de los sistemas de información geográfica (SIG) tiene una larga tradición de uso para la gestión y análisis de datos espaciales y es bien conocido entre los geo-especialistas: geógrafos, cartógrafos, topógrafos, entre otros. En particular, este tipo de sistemas proporciona muchas funcionalidades para apoyar a los usuarios en las complejas tareas de análisis de datos espaciales, sin hacer hincapié en otros aspectos, por ejemplo en el rendimiento o el almacenamiento. La práctica general es desarrollar los repositorios de datos espaciales basándose en una arquitectura híbrida donde los datos espaciales y convencionales se almacenan por separado (Worboys&Duckham, 2004), por ejemplo, utilizando archivos *shape* (ESRI, 1998) para los datos representados en un modelo vectorial. La referencia al archivo *shape* debe realizarse considerando por lo menos dos archivos: uno para almacenar las geometrías de los objetos y el otro para la representación de datos convencionales del tema de interés. En muchas ocasiones estos datos convencionales se encuentran repetidos en otros archivos *shape*, por ejemplo, los archivos *shape* en Costa Rica (como clínicas, centros educativos, agencias bancarias, entre otros) contienen datos repetidos sobre los distritos, cantones y provincias. Esta situación conlleva a un problema de alta repetición de datos y falta de control de su consistencia.

Afortunadamente, la situación ha cambiado con la adición de extensiones espaciales a los Sistemas de Administración de Bases de Datos (SABD) objeto-relacionales. Estos sistemas permiten tener en el mismo repositorio un conjunto de grandes volúmenes de datos convencionales (por ejemplo, cadenas de caracteres, numéricos, fechas) y no convencionales (por ejemplo, imágenes, geometrías). Además, proporcionan mecanismos de control de calidad de datos bien conocidos, por ejemplo, restricciones de integridad referencial, restricciones declarativas y dinámicas, así como otras nuevas características relacionadas con el manejo de datos espaciales, por ejemplo, índices espaciales, funciones espaciales y operaciones, o formatos y modelos para almacenar la colección de datos espaciales. Sin embargo, los SABD espaciales

(SABDE) no siempre se consideran como parte de los sistemas de información espacial o en las infraestructuras de datos espaciales, ya que representan una tecnología relativamente nueva y no bien conocida fuera de un pequeño círculo de implementadores. Además, la mayoría de los proyectos relacionados con los datos espaciales requiere conocimientos de geo-especialistas y por lo general ellos están más familiarizados con los SIG que SABDE. Adicionalmente, los SABDE requieren el conocimiento de base de datos que no siempre es una fortaleza de los geo-especialistas; algunos conceptos pueden ser complejos para ellos, tales como disparadores, restricciones de integridad referencial o distintos niveles de aislamiento para el control de concurrencia. Esta situación está cambiando lentamente debido a que los geo-especialistas se sienten presionados por la convergencia de los SIG y SABDE y deben capacitarse en las características y el uso de los bases de datos geo (*geodatabases*) (ESRI, 2012; Yeung y Hall, 2007).

En la actualidad, existen diferentes SABDE que pueden ser considerados para el almacenamiento y manipulación de datos espaciales, por ejemplo, PostgreSQL/PostGIS, Oracle, SQL Server, entre otros. Sin embargo, no existe una evaluación sistemática que permita comparar estos SABDE, ni una guía que facilite la decisión para seleccionar algún sistema de acuerdo con las necesidades específicas, por ejemplo, factores económicos (software libre o comercial), cumplimiento de los estándares, características de almacenamiento y rendimiento, entre otros. Algunas fuentes informales en Internet se refieren a diferentes características de los SABDE; sin embargo, son opiniones de los profesionales con base en sus experiencias sobre el uso de los sistemas sin presentar una comparación más rigurosa. Por otro lado, los artículos académicos referentes a la evaluación de los SABDE son demasiado viejos (Oosterom, Quak y Tijssen, 2002; McGranghan, 1997) y no consideran las nuevas características de estos sistemas o se refieren principalmente a los sistemas de código abierto realizando la comparación basada en las descripciones generales de las funciones que estos sistemas poseen (Chen and Xie, 2008).

En este artículo se presenta una evaluación de cuatro SABDE: PostgreSQL8.4 con PostGIS

1.5, MySQL Server 5.1, Oracle 11g R2 Enterprise Edition y SQL Server 2008 R2. Se consideran varios aspectos: disponibilidad (productos gratuitos o comerciales), el cumplimiento de los estándares y normas de datos espaciales, así como almacenamiento y rendimiento de las consultas. La Sección 2 se refiere a la evaluación basada en los criterios referentes a los estándares usados para los datos espaciales. La Sección 3 describe los detalles del caso de estudio utilizado para la evaluación del almacenamiento y rendimiento de las consultas. La Sección 4 incluye las comparaciones relacionados con el almacenamiento requerido para los índices espaciales y relaciones con instancias espaciales, mientras que la Sección 5 compara los tiempos de ejecución de las consultas agrupándolas de acuerdo con diferentes características. En la Sección 6 se presenta un resumen de las evaluaciones y se mencionan otros aspectos importantes de considerar antes de elegir una SABDE. Por último, la Sección 7 presenta las conclusiones de este artículo.

2. EVALUACIÓN BASADA EN ESTÁNDARES

El Comité Técnico ISO 211 (ISO/TC211), la organización internacional de estándares, y el Consorcio Abierto Geo-espacial (OGC, *Open Geospatial Consortium*), un consorcio industrial, son los actores principales en el ámbito de la normalización y unificación de los conceptos relacionados con la información geográfica. ISO/TC211 está encargado de desarrollar una serie de normas denominada serie ISO19100 (ISO/TC211, 2009) que no se refiere únicamente a los datos espaciales, sino también a la información geográfica y geomática. Por otra parte, OGC proporciona normas a disposición del público en forma de especificaciones abstractas y de implementación (OGC, 2012). La especificación abstracta está diseñada para proporcionar un marco conceptual para el desarrollo de interfaces de datos espaciales y servicios; la especificación de implementación está dirigida a los profesionales y técnicos responsables del desarrollo de sistemas que manejan datos espaciales. Ambos grupos, ISO/TC211 y OGC, colaboran en la unificación

de las especificaciones y la minimización de la superposición de elementos, por ejemplo, la especificación abstracta de OGC se adapta a varias normas de ISO/TC211 (ISO19107 corresponde al tema 1 de OGC, ISO19119 es equivalente al tema 12 de OGC, entre otros).

Por otra parte, el estándar SQL es el estándar más ampliamente utilizado en el contexto de las bases de datos. El estándar SQL:1999 (ISO/ IEX 13249:1999) (Stolze, 2003) incluye por primera vez las referencias a los datos espaciales en la parte 3 denominada SQL/MM (SQL *Multi-Media*). Este estándar define cómo almacenar, recuperar y manipular datos espaciales utilizando SQL. El estándar fue originalmente derivado de la especificación OGC (OGC, 1999) con una nueva modificación presentada en 2011 (SQL/MM, 2011). El estándar SQL/MM:2011 se centra en mantener la alineación con los estándares ISO/TC211 y OGC.

Malinowski (2010) propuso diferentes criterios de evaluación basados en ISO/TC211, OGC y SQL. Estos criterios se refieren a los tipos de geometría y métodos, modelos de las colecciones de objetos espaciales, sistemas de referencia espacial, interoperabilidad y metadatos como se puede ver en la Figura 1. Los criterios generales presentados en la figura fueron refinados por los componentes más específicos del estándar SQL/MM (SQL/MM, 2011). Por ejemplo, los tipos de geometría incluyen elementos como puntos, curvas, polígonos, entre otros, que a la vez se especializan en otros subtipos, como línea recta o circular.

Con base en estos criterios se evaluaron cuatro SABDE: PostgreSQL 8.4 con PostGIS 1.5, MySQL Server 5.1, Oracle 11g R2 Enterprise Edition y SQL Server 2008 R2. Debido a que SQL/MM proporciona una gran variedad de métodos que pueden ser utilizados para diferentes tipos de datos espaciales, con el fin de facilitar el proceso de evaluación, primero se agruparon estos métodos de acuerdo con los diferentes aspectos como se puede ver en la Tabla 1: recuperación de las propiedades o características espaciales (por ejemplo, tipo de geometría, longitud, área), creación de una nueva geometría a partir de una existente (por ejemplo, búfer, centroide), creación de una nueva geometría a partir de varias geometrías (por ejemplo, unión,

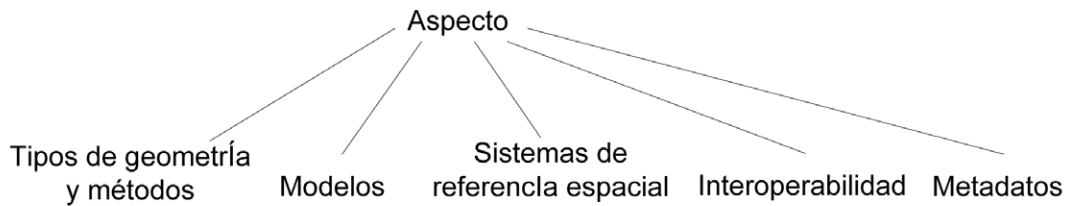


Figura 1. Criterios de evaluación.

intersección) y comparación de dos geometrías por medio de las relaciones topológicas (por ejemplo, tocar, disjunto). Además, los SABDE (excepto MySQL) proporcionan datos de tipo geometría y geografía que se utilizan para representar objetos en un plano o en la superficie de la Tierra, respectivamente. En este trabajo se utilizaron los tipos de datos de geometría en el espacio de dos dimensiones (2D) debido a que esta elección permite una comparación más amplia entre diferentes SABDE considerando que actualmente algunos sistemas (por ejemplo, PostGIS) han implementado más métodos para el tipo geometría que geografía.

La comparación se basa en el nuevo estándar SQL/MM:2011 y se realiza contando el número de elementos incluidos en el estándar SQL/MM:2011 y en cada uno de los SABDE de acuerdo con las categorías mencionadas anteriormente y presentes en la primera columna de la Tabla 1. Los resultados de la evaluación de cuatro SABDE se presentan en el Tabla 1.

Como se puede ver en la Tabla 1, actualmente PostGIS proporciona más funcionalidad relacionada con diferentes tipos de geometría y los métodos referentes a datos espaciales que otros SABDE evaluados, seguido por SQL Server y Oracle. Sin embargo, a pesar de que MySQL incluye varios métodos para relaciones topológicas, estos operan sobre los rectángulos delimitadores mínimos (MBR, *minimum bounding rectangle*) en lugar de las geometrías exactas. Oracle es el único que incluye un tipo de geometría nuevo llamado "sólido" que permite la creación y manipulación de polígonos en 3D. Además, Oracle de nuevo es el único que

proporciona tres modelos diferentes (espaguetis, topológico y de red) para las colecciones de los objetos espaciales, en contraste con otras SABDE que utilizan únicamente el modelo espaguetis, donde los objetos espaciales se definen de forma independiente. Todos los sistemas, excepto MySQL, permiten la consulta acerca de los metadatos relacionados con el ámbito espacial e incluyen un conjunto de métodos que garantiza la interoperabilidad, es decir, los métodos que transforman la representación de la geometría del formato propietario a *well-known text* (WKT), *well-knownbinary* (WKB), o *geographic markup language* (GML), y viceversa.

Algunos sistemas tienen métodos/operadores adicionales, por ejemplo, PostGIS incluye ST_AsEWKT para la transformación de la geometría a WKT extendido u Oracle tiene un operador para la búsqueda de vecinos más cercanos. Dado que el estándar SQL/MM no considera estos métodos, tampoco se incluyeron en el análisis, estableciendo el estándar SQL/MM como un punto de referencia para la evaluación.

3. CASO DE ESTUDIO

La evaluación del almacenamiento y desempeño de las consultas requiere la creación de un caso de estudio. Este caso se basa en los datos disponibles de los archivos *shape* referentes a Costa Rica, considerando diferentes tipos de geometrías, por ejemplo, puntos, líneas y polígonos. Estos datos se relacionan con diversos aspectos, por ejemplo, con la distribución administrativa, el medio ambiente, carreteras y ríos, factores

Tabla 1. Comparación de SABDE de acuerdo con el estándar SQL/MM.

Categoría	SQL/MM	PostGIS	MySQL	Oracle	SQL Server
Tipo de geometría	18	9	7	13	9
Métodos para recuperar propiedades	50	27	13	13	22
Métodos para crear una nueva geometría a partir de la otra	13	8	6	6	6
Métodos para crear una nueva geometría a partir de varias	4	4	0	4	4
Métodos para las relaciones topológicas	9	9	7	8	8
Métodos para la interpolación	6	6	6	6	6
Total de métodos	82	54	32	37	46
Modelos para la colección de objetos espaciales	3	1	1	3	1
Sistemas de referencias espacial soportado	No	Si	No	Si	Si
Administración de metadatos espaciales	Si	Si	No	Si	Si

económicos, entre otros. Los datos requieren una limpieza exhaustiva debido a su baja calidad y la falta de control sobre su consistencia. Por ejemplo, aunque Costa Rica tiene 475 distritos, uno de los archivos *shape* de distritos contiene 849 elementos. Esta situación se da debido a que los distritos compuestos por varias islas se representan tantas veces cuantas islas tienen; un caso extremo de esta representación es el distrito Sierpe representado en 61 elementos, donde el único atributo diferente es aquel que representa la geometría. Todos los demás datos convencionales (población, distribución de mujeres y hombres, entre otros) se repiten. Otros ejemplos que requieren una extensa limpieza se refieren a los archivos *shape* de cantones, debido a que algunas geometrías de sus correspondientes distritos están fuera de los límites. El conocimiento de diferentes mecanismos implementados en los SABDE facilitó el proceso de limpieza y transformación de los datos espaciales para obtener datos más exactos. Por ejemplo, se agruparon todos los registros referentes al mismo distrito creando las geometrías complejas, asegurándose de que cada distrito aparece solo una vez; se corrigieron las geometrías de distritos, cantones y provincias para asegurar la existencia de una jerarquía que representa adecuadamente la pertenencia de

cada distrito en su cantón y de cada cantón en su provincia; se invirtió el orden de especificación de las coordenadas en polígonos que representaban las provincias de Guanacaste y Puntarenas permitiendo evitar que se consideraran como “huecos”.

Se diseñó un almacén de datos espaciales usando el modelo conceptual MultiDim (Malinowski & Zimányi, 2008) y se realizó el mapeo al modelo objeto-relacional. Este almacén de datos se implementó en los cuatro SABDE.

Se utilizó la plataforma Windows 7 con el servidor Dell PowerEdge 1900, ACPI x64 con el procesador *quad-core* 2.66GHz y memoria de 4 GB como también el disco duro 3.5" SATA (7.2K RPM) con la capacidad de 250GB. No se modificaron ningunos parámetros para ajustar los SABDE a las particularidades de la aplicación, dejándolos con las opciones predeterminadas.

Para cargar datos, en primer lugar se utilizó una herramienta SIG que facilitó la exportación de los datos convencionales. Los datos espaciales se transformaron al formato WKT y luego se incluyeron en cada SABDE en su propio formato. Para la evaluación se utilizó un extracto de este almacén que incluye siete relaciones con datos espaciales y convencionales. Se seleccionaron las tablas que aseguraban el uso de diferentes

Tabla 2. Número de registros y tipos de geometría en las relaciones.

Nombre de la relación	Registros	Tipo de Geometría
Distrito	468	Multi-polígono
Cantón	81	Multi-polígono
Provincia	7	Multi-polígono
Reserva indígena	24	Polígono
Carretera	122967	Multi-línea
Río	31468	Multi-línea
Centro educativo	5056	Punto

tipos de geometría, que contenían número grande de registros o las que formaban jerarquías para evaluar la tarea de construir las geometrías a partir de otras geometrías, por ejemplo, crear cantones a partir de las geometrías de sus correspondientes distritos. La Tabla 2 especifica las relaciones seleccionadas con sus tipos de geometría y el número de registros que cada una contiene.

4. EVALUACIÓN DEL ALMACENAMIENTO

La evaluación de los SABDE, usando los criterios basados en estándares que se mostró en la Tabla 1, es útil para tener una visión general sobre las diferentes funcionalidades que estos sistemas ofrecen e indicar que tan cerca están a la especificación estándar. Por otra parte, el enfoque tradicional para la evaluación de los SABD es considerar el almacenamiento y desempeño de consultas. Estos dos aspectos también son importantes para los datos espaciales.

La evaluación de almacenamiento considerada en este proyecto se refiere al espacio en el disco necesario para los índices espaciales y las relaciones que contienen instancias espaciales. Las Figuras 2 y 3 muestran los resultados de estas comparaciones. Dado que las mediciones varían de 8kB a 33615kB para el almacenamiento de los índices y de 128kB a 75776kB para el almacenamiento de las relaciones, para representar los gráficos se utiliza la escala logarítmica con el objetivo de facilitar la comparación de la

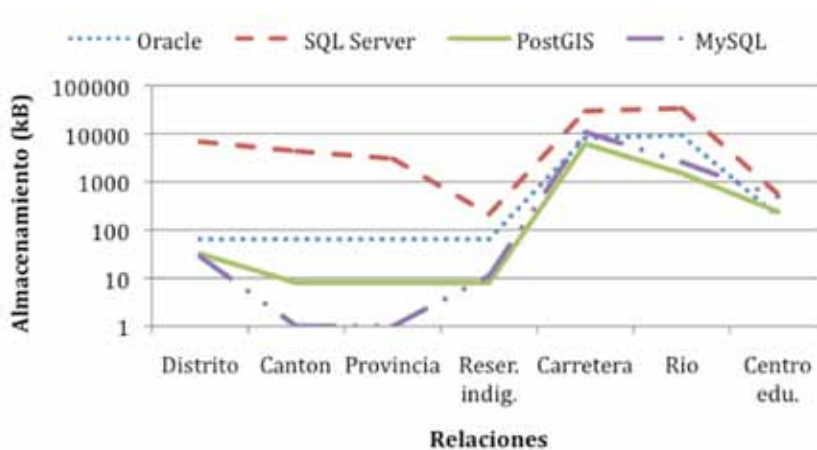
magnitud de almacenamiento requerido por cada uno de los SABDE. Para las geometrías que representan cantones se aplica la unión espacial de las geometrías de sus respectivos distritos. Una unión similar se aplica a las provincias considerando las geometrías de sus respectivos cantones. Sin embargo, MySQL no proporciona ningún método para la unión de las geometrías, por lo tanto, el almacenamiento de los índices y relaciones para los cantones y provincias no se pudo medir para MySQL.

Los SABDE evaluados utilizan diferentes tipos de estructuras de índices espaciales: PostGIS utiliza el índice de árbol R implementado sobre un árbol de búsqueda llamado GiST (*Generalized Search Tree*, Árbol Generalizado de Búsqueda), mientras que SQL Server aprovecha las ideas del árbol *quadtree* para la creación de etiquetas y los usa posteriormente en la estructura del árbol B+. Oracle y MySQL basan sus índices de la estructura del árbol R.

Como se puede ver en la Figura 2, SQL Server requiere la mayor cantidad del almacenamiento para los índices. El mejor indicador para el almacenamiento de índices se observa para PostGIS y sólo Oracle mejora esta característica cuando las geometrías están representadas como puntos. Sin embargo, existe un comportamiento no muy claro de Oracle y SQL Server para almacenar las líneas. A pesar de que la relación Río tiene casi cuatro veces menos registros que la relación Carretera, se requiere más espacio de almacenamiento para los índices. Se puede observar que PostGIS y MySQL difieren de este comportamiento.

Nombre de la relación	Oracle	SQL Server	PostGIS	MySQL
Distrito	64	6826	32	28
Cantón	64	4342	8	--
Provincia	64	3024	8	--
Reserva indígena	64	212	8	11
Carretera	8192	29433	6136	10739
Río	9216	33615	1504	2564
Centro educativo	192	553	240	488

a)



b)

Figura 2. Almacenamiento de índices espaciales (kB): a) valores exactos y b) representación gráfica.

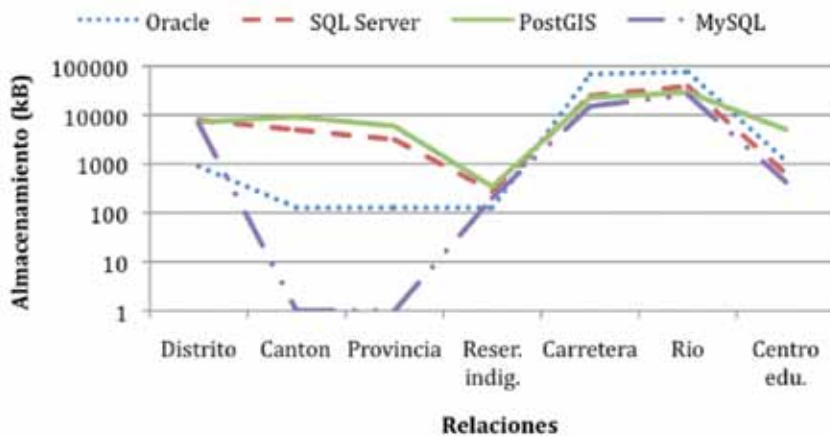
La Figura 3 muestra que PostGIS necesita más capacidad de disco que otros SABDE para almacenar relaciones con las geometrías de polígonos y puntos; sin embargo, requiere menos espacio que Oracle y SQL Server cuando las instancias espaciales están representadas por líneas. Por otra parte, Oracle ocupa menos espacio de almacenamiento para los polígonos que otros SABDE; sin embargo, este comportamiento cambia y da a Oracle el último lugar cuando las geometrías están representadas por líneas. MySQL utiliza menos espacio de almacenamiento para las tablas con puntos y líneas.

5. EVALUACIÓN DEL DESEMPEÑO

La evaluación del rendimiento incluye el tiempo para la creación de índices espaciales. La existencia de índices es importante para mejorar el desempeño de las consultas, especialmente para los datos espaciales debido a su gran volumen. Además, los índices se crean sobre los MBR (solo se guardan las coordenadas de dos puntos) y permiten la ejecución de consultas en dos fases: en la primera fase se seleccionan las geometrías "candidatas" considerando sus MBR; en la segunda fase se opera con las geometrías exactas de un conjunto limitado

Nombre de la relación	Oracle	SQL Server	PostGIS	MySQL
Distrito	896	7960	7264	6661
Cantón	128	4984	9200	--
Provincia	128	3152	5968	--
Reserva indígena	128	280	344	203
Carretera	68608	24848	22528	15155
Río	75776	38736	29225	25782
Centro educativo	1152	640	5056	431

a)



b)

Figura 3. Almacenamiento de relaciones con las instancias espaciales (kB): a) valores exactos y b) representación gráfica.

seleccionado en el paso 1, evitando de esta forma las comparaciones muy costosas de todas las geometrías.

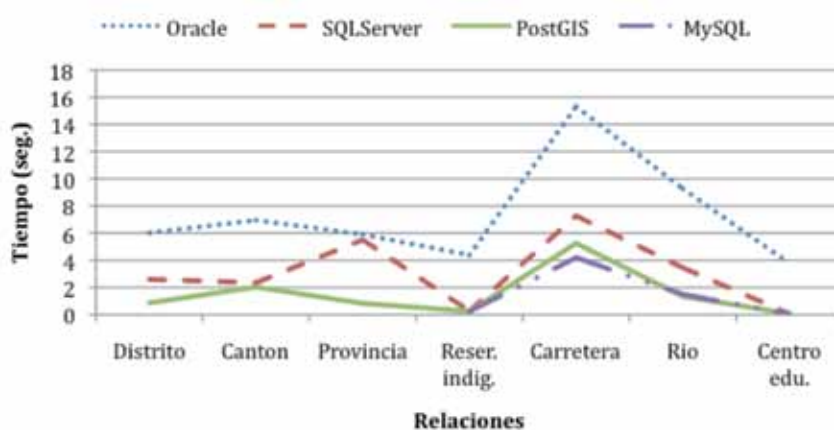
SQL/MM proporciona una gran variedad de métodos que pueden ser aplicados para diferentes tipos de datos espaciales (SQL/MM, 2011). Se utiliza la terminología de SQL/MM para especificar los métodos evaluados, aunque algunos SABDE pueden llamarlos de manera diferente. Para facilitar el proceso de evaluación se especifican los mismos grupos de consultas como los mencionados en la Sección 2: recuperación de las propiedades o características espaciales, creación de una nueva geometría a

partir de la otra ya existente o a partir de varias geometrías y comparación de dos geometrías con respecto a las relaciones topológicas. Estos métodos se ejecutan para diferentes tipos de geometrías en tres pruebas (arranque en frío) y los resultados obtenidos se promedian.

Los SABDE usan diferentes formas de implementación de los métodos para la manipulación de datos espaciales. Por ejemplo, Oracle distingue tres diferentes grupos: (1) métodos asociados con las instancias espaciales para recuperar la información acerca de un objeto espacial, (2) los operadores espaciales utilizados en la cláusula *where* de SQL, los cuales requieren la existencia de los índices

Nombre de la relación	Oracle	SQL Server	PostGIS	MySQL
Distrito	896	7960	7264	6661
Cantón	128	4984	9200	--
Provincia	128	3152	5968	--
Reserva indígena	128	280	344	203
Carretera	68608	24848	22528	15155
Río	75776	38736	29225	25782
Centro educativo	1152	640	5056	431

a)



b)

Figura 4. Tiempo de ejecución para la creación de índices espaciales (segundos): a) valores exactos y b) representación gráfica.

espaciales y (3) los procedimientos y funciones de PL/SQL que no utilizan índices y se incluyen en el paquete llamado `SDO_GEOM`. Por otro lado, la mayoría de los métodos de PostGIS, con la excepción de algunos de ellos (por ejemplo, `ST_Distance`) requiere de los índices espaciales para acelerar la ejecución de la consulta y el optimizador de consultas es responsable de tomarlos en consideración. Sin embargo, de acuerdo con los desarrolladores de PostGIS, los índices no siempre se consideran en el plan de ejecución, realizando en cambio un escaneo secuencial de toda la tabla (PostgreSQL, 2011). En cambio, SQL Server utiliza los índices espaciales en la cláusula *where* sólo para las relaciones topológicas y para el método de `STDistance`.

En el proceso de evaluación del desempeño de consultas, se excluyó MySQL debido a que

implementa muy pocos métodos y los métodos para las relaciones topológicas operan sobre MBR, lo que ofrece sólo resultados aproximados y facilita la ejecución considerando el hecho de que sólo dos puntos se necesitan recuperar para representar cada geometría. Por lo tanto, se valoró que los resultados dados por MySQL no deben ser comparados con otros SABDE. La única referencia a MySQL se hace cuando se evalúa el tiempo requerido para la creación de índices.

5.1 Creación de índices espaciales

Los tiempos necesarios para la creación de índices espaciales se muestran en la Figura 4 para las siete relaciones especificadas en Tabla 2. Como

ya se ha indicado en el apartado anterior, MySQL no proporciona ningún método para la unión de las geometrías; por lo tanto, los tiempos para la creación de índices de las relaciones que representan los cantones y provincias no se muestran en la figura. Además, el tiempo para la creación de índices para la relación de distritos es tan pequeño (0.099 segundos) que no se percibe en el gráfico tampoco. (Figura 4b)

Como se puede ver en la Figura 4, Oracle necesita más tiempo para crear los índices que otros SABDE, seguido, en la mayoría de los casos, por el SQL Server. Además, es interesante observar que, para SQL Server, aunque las relaciones de distrito, cantón y provincia contienen las geometrías compuestas por polígonos con un número decreciente de la cantidad de registros de 468 a 7, los tiempos de ejecución aumentan de 2,6 a 5,5 segundos (más que doble), mientras que otros SABDE muestran un comportamiento diferente. Por otro lado, el comportamiento "normal" se observa para las líneas (las relaciones de Carretera y Río) para los cuatro SABDE donde la disminución en el número de registros requiere menos tiempo para crear los índices. PostGIS logra los mejores tiempos de ejecución considerando todos los tipos de geometría.

5.2 Recuperación de propiedades de datos espaciales

Los métodos para la recuperación de las propiedades de datos espaciales permiten a los usuarios consultar diferentes características de estos datos antes de analizarlos o integrarlos. Se seleccionaron tres métodos: ST_Dimension que devuelve el número de dimensiones utilizado para la representación de la geometría, es decir, el número 2 debido a que se considera el espacio de dos dimensiones, ST_GeometryType que despliega el nombre del tipo de geometría utilizada para representar instancias espaciales y ST_IsEmpty que devuelve verdadero (*true*) si la geometría es una geometría vacía. Estos métodos se aplicaron a diferentes tipos de geometrías y relaciones, como se puede ver en la Tabla 3.

La Figura 5 muestra los resultados de ejecutar las consultas con los métodos especificados en el Tabla 3. Oracle no implementa el método ST_IsEmpty (el valor faltante para la consulta Q7). Por otra parte, el tiempo de ejecución de Oracle

para la relación Carretera es tan alto (alrededor de 8 minutos) en comparación con el menor tiempo de PostGIS para la relación Distrito (alrededor de 47 segundos) que se utilizó una escala logarítmica para el gráfico (ver Figura 5b). Esto permite apreciar mejor en la forma gráfica la magnitud de diferentes tiempos de ejecución que representan las SABDE. Como se puede ver en la figura, Oracle necesita el mayor tiempo de ejecución seguido por SQL Server y PostGIS en todas las consultas realizadas.

5.3 Recuperación de las características espaciales

Otro grupo de consultas recupera las características espaciales. Se utilizaron tres métodos: para calcular la longitud de las líneas (ST_Length), así como el área (ST_Area) y el perímetro (ST_Perimeter) de los (multi-) polígonos, como se puede ver en la Tabla 4.

La Figura 6 a) incluye los tiempos de ejecución exactos, mientras que la Figura 6 b) muestra los gráficos con los tiempos de ejecución presentados en la escala logarítmica. De forma similar al caso anterior, como se muestra en la figura, PostGIS tiene un tiempo de ejecución mejor que Oracle y SQL Server. Una vez más, el peor rendimiento de estos métodos lo tiene Oracle.

5.4 Generación de una nueva geometría a partir de una existente.

El siguiente grupo de consultas permite a los usuarios generar una nueva geometría a partir de una ya existente. Se aplicaron cinco diferentes métodos: uno de ellos genera un punto (ST_Centroid), dos métodos crean un polígono (ST_ConvexHull, ST_Buffer), y dos producen resultados diferentes dependiendo del tipo de geometría aplicado (ST_Boundary, ST_Envelope). La Tabla 5 especifica los métodos usados y las relaciones a las cuales se aplicaron, mientras que la Figura 7 muestra los resultados obtenidos.

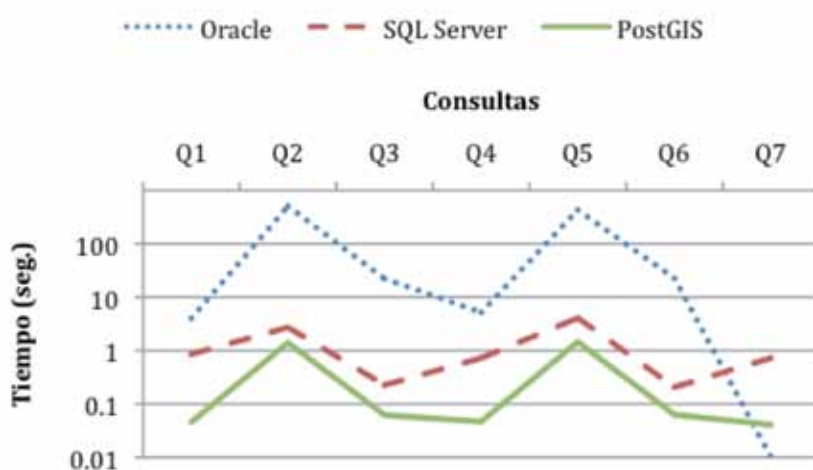
Oracle implementa todos los métodos especificados en esta sección como paquetes. No implementa el método ST_Boundary (falta de valor en la Figura 7 para la consulta Q1). Además,

Tabla 3. Métodos y relaciones usados para la Figura 5.

Consulta	Método	Relación
Q1	ST_Dimension	Distrito
Q2	ST_Dimension	Carretera
Q3	ST_Dimension	Centro Educ.
Q4	ST_GeometryType	Distrito
Q5	ST_GeometryType	Carretera
Q6	ST_GeometryType	Centro Educ.
Q7	ST_IsEmpty	Distrito

Consulta	Oracle	SQL Server	PostGIS
Q1	4.02	0.857	0.046
Q2	505.045	2.714	1.404
Q3	22.010	0.226	0.062
Q4	5.190	0.726	0.047
Q5	427.360	4.069	1.450
Q6	22.520	0.207	0.063
Q7	--	0.728	0.041

a)



b)

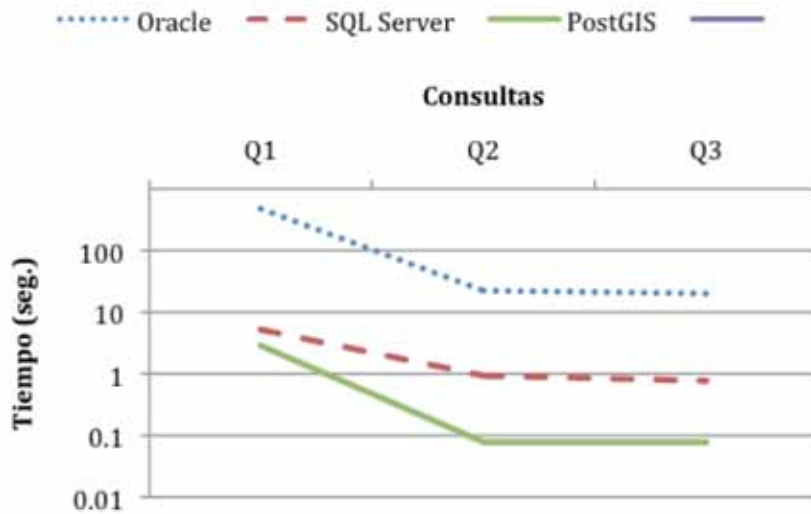
Figura 5. El tiempo de ejecución para recuperar las propiedades de datos espaciales (segundos): a) valores exactos y b) representación gráfica.

Tabla 4. Métodos y relaciones usadas para la Figura 6.

Consulta	Método	Relación
Q1	ST_Length	Carretera
Q2	ST_Area	Distrito
Q3	ST_Perimeter	Distrito

Consulta	Oracle	SQL Server	PostGIS
Q1	475.170	5.205	2.870
Q2	22.290	0.932	0.078
Q3	20.076	0.771	0.078

a)



b)

Figura 6. Tiempo de ejecución para la recuperación de las características espaciales (segundos): a) valores exactos y b) representación gráfica.

Oracle utiliza el método de ST_ConvexHull solamente para las geometrías diferentes de puntos o líneas rectas, mientras PostGIS y SQL Server no incluyen esta restricción devolviendo la misma geometría para los tipos de geometría mencionados

anteriormente. Por lo tanto, para facilitar la comparación, se utilizó este método para multi-polígonos (Distrito) y líneas (Carreteras) como se puede ver en la Tabla 5. El método ST_Centroid se puede utilizar para los puntos (representados como

multi-puntos), líneas y polígonos en PostGIS; sin embargo, Oracle sólo utiliza este método para multi-puntos y polígonos, mientras SQL Server sólo para los polígonos, devolviendo los valores nulos en el otro caso. Todos los SABDE devuelven el mismo tipo de geometría cuando el método ST_Envelope se aplica a puntos y líneas rectas.

Oracle ejecuta más lentamente todas las consultas especificadas en comparación con PostGIS y SQL Server. La diferencia entre Oracle y PostGIS es significativa para todos los métodos, donde se nota un aumento del tiempo de ejecución en aproximadamente 1150 veces para el método ST_ConvexHull aplicado a la relación Carretera (Q6), como también para el método ST_Buffer utilizado con la relación Distrito (Q7); esta diferencia es menor entre Oracle y SQL Server, dando incremento de 560 y 425 veces para las consultas mencionadas anteriormente (Q6 y Q7, respectivamente). Un comportamiento anormal se puede observar para el método ST_Envelope (Q4) donde Oracle lo ejecuta 3500 veces más lento que SQL Server.

La comparación de los tiempos de ejecución entre SQL Server y PostGIS presenta diferencias menores. Como se puede ver en la Figura 7, por un lado, la ventaja la toma PostGIS sobre SQL Server con los tiempos de ejecución que oscilan entre los 23 veces menor para la consulta

Q3 y 1,2 veces menor para la consulta Q5. Por otro lado, SQL Server mejora sus tiempos de ejecución sobre PostGIS en 9 y 1,4 veces para las consultas Q4 y Q9, respectivamente. Además, en tres casos (Q3, Q7 y Q10) de cuatro, cuando los métodos se aplicaron a la relación Distrito, SQL Server ejecutó estas consultas más lentamente que PostGIS.

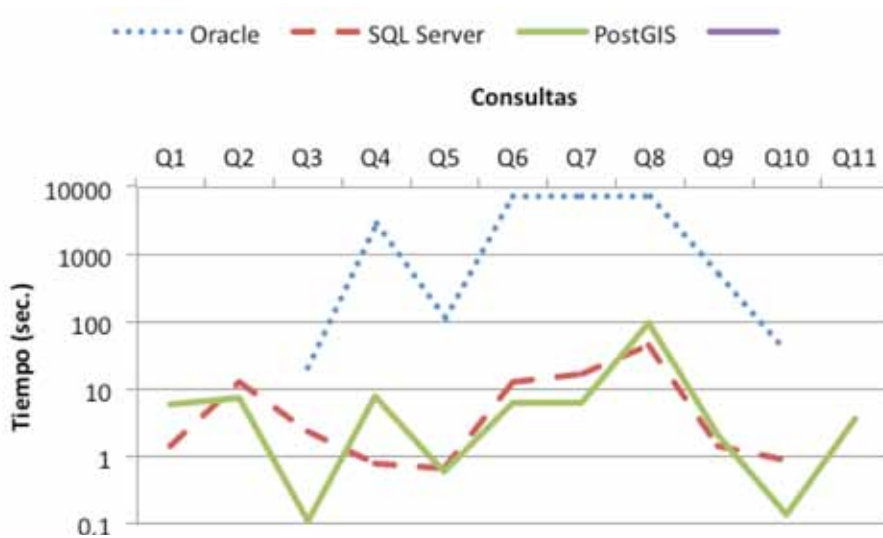
Otra comparación entre los SABDE se muestra en la Tabla 6 donde se agruparon los métodos de acuerdo con los tipos de geometría y, dentro de cada grupo, se ordenaron los métodos del más rápido al más lento con respecto a los tiempos de ejecución para los SABDE dados. Todos los sistemas producen los peores tiempos de ejecución para el método ST_Buffer, independientemente del tipo de geometría. Para multi-polígonos, aunque PostGIS utiliza menos tiempo que Oracle, representa una clasificación de los tiempos de ejecución muy similar a la de Oracle, mientras que SQL Server se diferencia de estos dos sistemas en los métodos ST_Envelope y ST_ConvexHull. El método más rápido para Oracle requiere 200 veces menos tiempo que el método más lento, mientras que esta diferencia es menor para SQL Server (25 veces) y PostGIS (59 veces). Para las líneas, la clasificación es similar para los tres sistemas evaluados, debido a

Tabla 5. Métodos y relaciones para la Figura 6.

Consulta	Método	Relación
Q1	ST_Boundary	Distrito
Q2	ST_Boundary	Carretera
Q3	ST_Envelope	Distrito
Q4	ST_Envelope	Carretera
Q5	ST_ConvexHull	Distrito
Q6	ST_ConvexHull	Carretera
Q7	ST_Buffer	Distrito
Q8	ST_Buffer	Carretera
Q9	ST_Buffer	Centro educ.
Q10	ST_Centroid	Distrito
Q11	ST_Centroid	Carretera

Consulta	Oracle	SQL Server	PostGIS
Q1	--	1.515	6.069
Q2	--	13.022	7.472
Q3	21.740	2.496	0.109
Q4	2928.02	0.822	7.987
Q5	116.042	0.703	0.593
Q6	7390.03	13.192	6.318
Q7	7454.030	17.601	6.442
Q8	7338.01	46.493	97.298
Q9	522.019	1.513	2.106
Q10	37.036	0.926	0.141
Q11	--	--	3.588

a)



b)

Figura 7. Tiempo de ejecución para generar una geometría a partir de la otra (segundos):
a) valores exactos y b) representación gráfica.

que las diferencias de tiempos de ejecución entre los métodos ST_ConvexHull, ST_Boundary y ST_Envelope para PostGIS son insignificantes. Por otra parte, la relación entre los tiempos

de ejecución más lentos y más rápidos para esta categoría de métodos es de 2,5 veces para Oracle, 56 veces para SQL Server y 27 veces para PostGIS.

Tabla 6. Métodos ordenados desde el más rápido hasta el más lento para cada grupo de geometría.

Tipo de geometría	Oracle	SQL Server	PostGIS
Multi-polígono (Distrito)	ST_Envelope (Q3)	ST_ConvexHull (Q5)	ST_Envelope (Q3)
	ST_Centroid (Q10)	ST_Centroid (Q10)	ST_Centroid (Q10)
	ST_ConvexHull (Q5)	ST_Boundary (Q1)	ST_ConvexHull (Q5)
	ST_Buffer (Q7)	ST_Envelope (Q3)	ST_Boundary (Q1)
		ST_Buffer (Q7)	ST_Buffer (Q7)
Multi-línea (Carretera)	ST_Envelope (Q4)	ST_Envelope (Q4)	ST_Centroid (Q11)
	ST_ConvexHull (Q6)	ST_Boundary (Q2)	ST_ConvexHull (Q6)
	ST_Buffer (Q8)	ST_ConvexHull (Q6)	ST_Boundary (Q2)
		ST_Buffer (Q8)	ST_Envelope (Q4)
			ST_Buffer (Q8)

Tabla 7. Métodos y relaciones usadas para la Figura 8.

Consulta	Método	Relación
Q1	ST_Union	Distrito
Q2	ST_Difference	Distrito, Reserva Ind.
Q3	ST_Intersection	Distrito, Reserva Ind.

5.5 Generación de una nueva geometría a partir de varias

Para el proceso de generación de una nueva geometría a partir de otras geometrías, se utilizaron tres métodos: ST_Union, ST_Difference y ST_Intersection con las relaciones indicadas en el la Tabla 7.

Al igual que en la sección anterior, la Figura 8a) presenta los tiempos de ejecución exactos, mientras que la Figura 8b) presenta los tiempos de ejecución en la escala logarítmica para Oracle, SQL Server y PostGIS. Como se puede ver, Oracle tiene el peor tiempo de ejecución de los tres SABDE. PostGIS es 8,5 veces más rápido que SQL Server para el método ST_Union (Q1). La situación contraria ocurre con los métodos ST_Difference (Q2) y ST_Intersection (Q3), cuando SQL Server es,

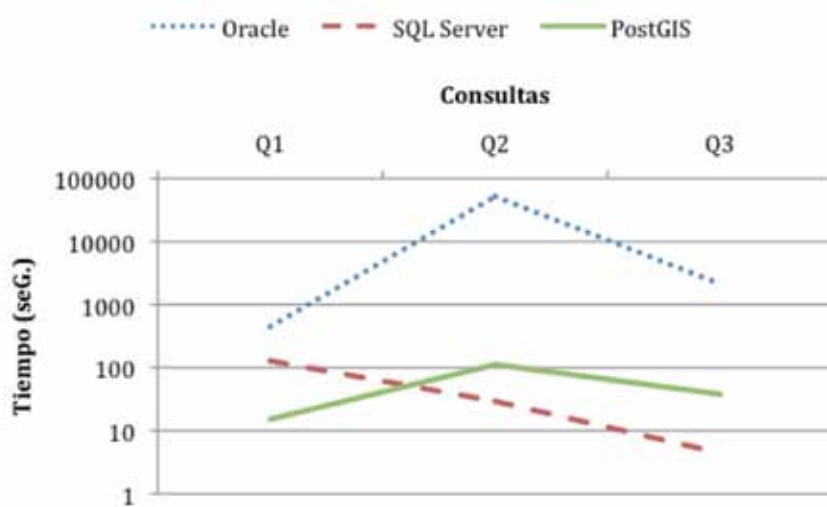
respectivamente, 3,8 y 8,5 veces más rápido que PostGIS. Además, SQL Server muestra un comportamiento diferente de PostGIS y Oracle para el método ST_Difference y es casi 1800 veces más rápido que Oracle, que tarda más de 14 horas para completar su ejecución.

5.6 Comparación de geometrías por medio de relaciones topológicas

Las relaciones topológicas son métodos muy importantes para consultar datos espaciales y pueden ser costosas debido a las comparaciones requeridas de dos geometrías que pueden ser muy complejas. La Tabla 8 indica los métodos y relaciones utilizados para cada consulta. Para Oracle se usaron los operadores, debido a que, de acuerdo con Oracle, estos funcionan mejor.

Consulta	Oracle	SQL Server	PostGIS
Q1	447.000	128.000	15.179
Q2	51894.000	29.278	111.977
Q3	2126.080	4.426	37.955

a)



b)

Figura 8. Los tiempos de ejecución para generar una geometría a partir de varias (segundos): a) valores exactos y b) representación gráfica.

Como se puede ver a partir de los resultados mostrados en la Figura 9, PostGIS ejecuta las consultas más rápido que otros SABDE, con la excepción del método `ST_Relate` (Q2) donde Oracle y SQL Server toman una pequeña ventaja. La situación es inversa cuando se ejecuta la consulta con el método `ST_Crosses` (Q3), donde PostGIS es 4 veces más rápido que Oracle. El método más rápido para todos los sistemas es `ST_Equal` (Q1) y su tiempo de ejecución oscila entre 0,453 seg. (PostGIS) y 3,096 seg. (Oracle). El otro método rápido para SQL Server y PostGIS es el `ST_Relate` (Q6) y para Oracle el método `ST_Touches` (Q4). Se puede hacer una comparación interesante relacionada con el uso de la relación topológica *within*.

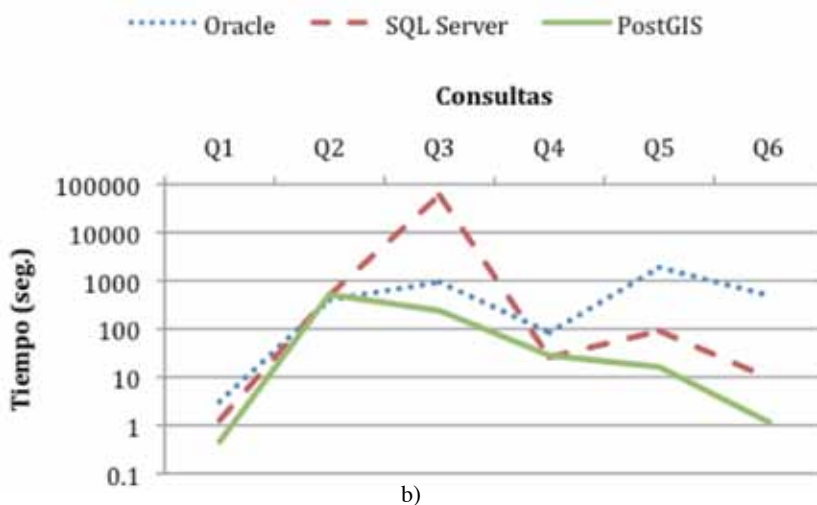
Esta relación topológica se puede obtener por medio de dos métodos: `ST_Relate` (Q2) y `ST_Within` (Q6). Como puede verse en la Figura 9 existen diferencias considerables en los tiempos de ejecución de estos dos métodos para SQL Server y PostGIS. El método `ST_Relate` (Q2) se ejecuta 53 veces más lento en SQL Server y 450 veces más lento en PostGIS en comparación con el método `ST_Within`. Un comportamiento diferente se puede notar para Oracle, cuando el método `ST_Within` se ejecuta 1,2 veces más lento que el método `ST_Relate`. En general, no existe un patrón regular en los tiempos de ejecución utilizando diferentes relaciones topológicas y, se puede decir que PostGIS tiene una ventaja en comparación con los otros SABDE.

Tabla 8. Métodos y relaciones para la Figura 9.

Consulta	Método	Relación
Q1	ST_Equal	Distrito, Reserva Ind.
Q2	ST_Relate	Distrito, Centro Educ.
Q3	ST_Crosses	Carretera, Río
Q4	ST_Touches	Cantón, Distrito
Q5	ST_Intersects	Distrito, Carretera
Q6	ST_Within	Distrito, Centro Educ.

Consulta	Oracle	SQL Server	PostGIS
Q1	3.096	1.267	0.453
Q2	400.056	504.122	518.467
Q3	937.021	59677.43	238.649
Q4	83.990	25.826	28.174
Q5	1884.700	90.802	16.193
Q6	494.005	9.482	1.155
Q1	3.096	1.267	0.453

a)



b)

Figura 9. Los tiempos de ejecución para la comparación de dos geometrías por medio de relaciones topológicas (segundos): a) valores exactos y b) representación gráfica.

6. RESUMEN Y OTROS ASPECTOS

Los tiempos de ejecución de los diferentes métodos que pertenecen a grupos establecidos para esta investigación muestran que, por un lado, Oracle es un SABDE más lento que otros con sólo unas pocas excepciones y, por otro lado, PostGIS proporciona el mejor rendimiento en la mayoría de los experimentos. Además, cuando SQL Server ejecuta el método de la relación topológica ST_Crosses y Oracle el método ST_Difference los tiempos de ejecución son muy elevados (más de 16 y 14 horas, respectivamente) en comparación con los otros SABDE. Al mismo tiempo, los SABDE pueden proporcionar diferentes opciones para expresar la misma consulta, por ejemplo, para las relaciones topológicas; estas variaciones pueden conducir a tener muy variados tiempos de ejecución, como se pudo notar para los métodos ST_Within y ST_Relate. Igualmente, aunque algunos sistemas pueden incluir una gran variedad de métodos, estos métodos se pueden aplicar sobre los MBR (por ejemplo, relaciones topológicas en MySQL), no siempre utilizan los índices espaciales (por ejemplo, PostGIS), o pueden ser implementados "fuera" del motor de SABDE como subprogramas (por ejemplo, Oracle).

Aparte del almacenamiento y los tiempos de ejecución, se puede considerar otros aspectos al comparar diferentes SABDE:

- **Complejidad sintáctica:** PostGIS y SQL Server utilizan las notaciones (y nombres) propuestos por el estándar SQL/MM basado en objetos que invocan sus métodos, en contraste con Oracle y MySQL que usan un enfoque más cercano a la programación estructurada, donde los objetos espaciales se utilizan como parámetros para las funciones u operadores. Además, Oracle utiliza la sintaxis más compleja para la inserción de diferentes tipos de geometrías que requiere el conocimiento de muchos parámetros y características de las bases de datos objeto-relacionales.
- **El conocimiento especializado sobre las características de los SABDE:** Todos los sistemas permiten la implementación de diferentes tipos de geometría; sin embargo, Oracle requiere adicionalmente la actualización de los metadatos relacionados con atributos

espaciales, mientras que otros sistemas lo pueden manejar de forma transparente a los usuarios. Además, la definición de los índices espaciales en SQL Server requiere el establecimiento de un marco de referencia espacial para los índices (sólo una parte de las instancias espaciales puede ser indexada) y el conocimiento acerca de los diferentes niveles de las precisiones del árbol *quadtree*.

- **Problemas o limitaciones del producto:** Muchos problemas relacionados con las diferentes SABDE se pueden encontrar en diferentes grupos de discusión en la Internet; se mencionará solo algunos de ellos. Por ejemplo, SQL Server requiere algunas correcciones de las geometrías antes de su manipulación (Aitchison, 2009). Este proceso de limpieza, especialmente para los datos procedentes de los archivos *shape*, puede ser complejo y puede incluir la aplicación de diferentes métodos para corregir las geometrías que no son simples, para asegurar la orientación correcta del anillo, o para "suavizar" los puntos (Aitchison, 2009). Adicionalmente, SQL Server con su versión 2008 sólo permite los tipos de datos geográficos que no se expanden más que sobre un hemisferio (Aitchison, 2009). Para PostGIS se presenta otra situación, pues utiliza las tablas llamadas TOAST para almacenar valores grandes, tales como coordenadas de las geometrías; este almacenamiento tiene un problema:

"El problema aparece cuando se tiene una relación con geometrías grandes, pero no con demasiados registros (como una tabla que contiene los límites de todos los países europeos en alta resolución). La tabla por sí sola es pequeña, pero utiliza mucho espacio de TOAST. En nuestro caso de ejemplo, la tabla en sí tenía alrededor de 80 registros y utilizaba sólo 3 páginas de datos, pero esta tabla en el TOAST necesitaba 8225 páginas." (PostgreSQL: 49, 2011)

Otro problema fue descubierto accidentalmente con Oracle cuando la operación `sdo_area` se aplicó a un polígono mal definido. Los estudiantes durante la práctica crearon un polígono usando la línea de límite no cerrada.

Como Oracle no comprueba esta condición, aplicó el operador `sdo_area` reportando un resultado de 1 unidad en lugar de 2. Se pudo encontrar este error debido a que las geometrías definidas fueron muy simples. Sin embargo, debe realizarse una revisión adicional para asegurar la exactitud de los resultados y establecer las restricciones adicionales a fin de evitar definiciones incorrectas de las geometrías.

- **La existencia de características adicionales:** Los sistemas evaluados utilizan la representación vectorial basada en el modelo espagueti donde todas las geometrías se definen por separado. Sin embargo, en algunas ocasiones los usuarios pueden necesitar los modelos topológicos o de red (Kothuri, Godfrind y Beinat, 2007). Actualmente, sólo Oracle proporciona estos modelos. Otras características incluidas por Oracle y carentes en otros sistemas son las operaciones más avanzadas, como por ejemplo, los vecinos más cercanos, unión espacial, geo-codificación, o el modelo raster (Oracle, 2009). La última característica está ya desarrollada también para PostGIS (PostgreSQL, 2011).
- **Manejo de tipos de datos geográficos:** Oracle y SQL Server permiten definir y aplicar los mismos métodos para los tipos de datos definidos como geometría (en plano) y geografía (definidos sobre la superficie de la Tierra). Sin embargo, la situación es diferente para PostGIS, el cual cuenta con una gran variedad de métodos para el tipo geometría y con un número más limitado de métodos para el tipo de datos geografía.
- **Facilidades para la incorporación con los SIG:** Como se indicó anteriormente, las bases de datos espaciales proporcionan facilidades en almacenamiento junto con los controles declarativos y dinámicos sobre los datos. Estas características son inexistentes en los SIG tradicionales, especialmente cuando operan sobre los archivos *shape*. Sin embargo, los SIG son una mejor opción para el análisis de los datos y la visualización. Por lo tanto, es importante considerar si la integración entre los SABDE y SIG es un proceso simple. La mayoría de los SIG proporcionan conectividad a PostGIS,

MySQL u Oracle. Sin embargo, la extensión espacial de SQL Server es relativamente nueva y aparece por primera vez en la edición de 2008; como consecuencia, todavía faltan los controladores que permiten importar/exportar datos entre SQL Server y las herramientas SIG, aunque existe una clara tendencia a lograr esta conectividad.

7. CONCLUSIONES

- Los repositorios tradicionales de datos espaciales basados en archivos shape no proporcionan las características requeridos para la creación de los controles necesarios con el objetivo de mejorar la calidad y exactitud de este tipo de datos. Por lo tanto, se deberían analizar las diferentes alternativas para la creación de sistemas de información espacial, como por ejemplo los sistemas de administración de base de datos (SABD) con extensiones espaciales. Este tipo de sistemas proporciona características bien conocidas de controles declarativos y dinámicos para mejorar la consistencia de los datos. Sin embargo, los SABD espaciales (SABDE) no siempre se consideran como una opción del almacenamiento de datos espaciales, debido a que es una tecnología relativamente nueva y poco conocida entre los geo-especialistas quienes en muchos casos están a cargo de la creación de sistemas de información espacial.
- Este trabajo presentó la evaluación de cuatro SABDE: PostgreSQL/PostGIS, MySQL, Oracle y SQL Server considerando el cumplimiento del estándar SQL/MM:2011 y los requisitos de almacenamiento. Además, se realizaron diversos experimentos para comparar el desempeño de las consultas, excluyendo MySQL de esta última evaluación, debido a que no implementa muchos métodos y algunos de ellos operan sobre los MBR, dando resultados aproximados.
- En general, la evaluación muestra claramente que PostGIS es la mejor opción si se requiere manipular los datos vectoriales en el modelo espaguetis. Este sistema incluye una cantidad mayor de diferentes

tipos de geometría y métodos, tiene un mejor indicador de almacenamiento (a pesar de algunos problemas) y, en muchos casos, se caracteriza por ofrecer el mejor tiempo de ejecución para diferentes tipos de consultas. Además, al ser un SABDE libre y de código abierto no requiere inversión en su adquisición. Por otra parte, Oracle y SQL Server 2008 proporcionan un rico conjunto de tipos espaciales y métodos; sin embargo, SQL Server requiere más espacio de almacenamiento para los índices espaciales que Oracle. Por otro lado, en la mayoría de los casos, Oracle tiene el peor tiempo de ejecución para diferentes tipos de consultas.

- Los resultados de este trabajo pueden ser útiles para diferentes tipos de usuarios en la elección de la alternativa de almacenamiento para la implementación de sistemas de información espacial o infraestructura de datos espaciales. En particular, estos SABDE no sólo proporcionan mejores capacidades de almacenamiento en un entorno integrado, sino que también permiten implementar mecanismos de control de datos y, como consecuencia, mejorar la calidad de estos datos. De esta forma, los usuarios pueden usarlos en los procesos de toma de decisiones considerando no solo los SIG, sino también otros tipos de herramientas como lo son OLAP (*On-Line Analytical Processing*, Procesamiento Analítico en Línea) espacial (Gomez, Vaisman y Zich, 2008) y minería de datos espaciales (Lin *et al.*, 2011).

8. REFERENCIAS BIBLIOGRÁFICAS

- Aitchison, A. (2009). *Beginning spatial with SQL Server 2008*. Apress.
- Chen, R. & Xie, J. (2008). Open sources databases and their spatial extensions. En G. Brent Hall & M. Leahy (eds.) *Open Sources Approaches in Spatial Data Handling*. Springer.
- ESRI. (1998). *Shapefile Technical Description*. Recuperado el 12 de agosto, 2010 de <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>.
- ESRI. (2012). *Geotababase: overview*. Recuperado el 12 de agosto, 2011 de <http://www.esri.com/software/arcgis/geodatabase/index.html>.
- Gomez, L., Vaisman, A. & Zich, A. (2008). Piet-QL: A Query Language for GIS-OLAP integration. *En Memorias de ACM SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems*.
- ISO/TC211. (2009). *Standards Guide: ISO/TC211 Geographic Information/Geomatics*. Recuperado el 15 de marzo, 2010 de http://www.isotc211.org/Outreach/ISO_TC_211_Standards_Guide.pdf.
- Kothuri, R., Godfrind, A. & Beinat, E. (2007). *Oracle Spatial for Oracle Database 11g*. Apress.
- Lin, J., Chen, Ch., Wu, X., Wu, J. & Wang, W. (2011). GeoKSGrid: A Geographical Knowledge Grid with Functions of Spatial Data Mining and Spatial Decision. *En Memorias de IEEE Int. Conf. on Spatial Data Mining and Geographical Knowledge Services*.
- McGranagham, M. (1997). Spatial Data Models in Current Commercial RDBMS. *En Memorias de ACSM/ASPRS Annual Convention & Exposition*.
- Malinowski, E. (2010). Spatial Databases as a Storage Alternative for Spatial Information Systems. *En Memorias de 2010 Int. Conf. on Information and Knowledge Engineering*.
- Malinowski, E. & Zimányi, E. (2008). *Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications*. Springer.
- OGC. (2012). *OGC Standards and Supporting Documents*. Recuperado el 23 de mayo, 2011 de <http://www.opengeospatial.org/standards>.
- OGC. (1999). *OpenGIS simple features specification for SQL: Technical Report Revision 1.1*
- Oosterom, P., Quak, C. & Tijssen, T. (2002). Testing Current DBMS Products with Real Spatial Data. *En Memorias de Urban Data Management Symposium*.
- Oracle. (2009). *Oracle Spatial: Developer's Guide, 11g Release*. Recuperado el 23 de mayo 2011 de http://docs.oracle.com/cd/B28359_01/appdev.111/b28400/toc.htm.

- PostgreSQL. (2011). *Manual, version 1.5.1*. Recuperado el 12 de abril 2011 de <http://postgis.refractor.net/download/postgis-1.5.1.pdf>.
- SQL/MM. (2011). *ISO/IEC 13249-3:2011. SQL Multimedia and Application Packages - Part 3: Spatial*.
- Microsoft. (2008). *SQL Server Books On-line*. Recuperado el 13 de febrero de 2010 de <http://www.microsoft.com/en-us/download/details.aspx?id=1054>
- Stolze, K. (2003). The Standard to Manage Spatial Data in Relational Database Systems. En *Memorias de 10th Conference on Database Systems for Business, Technology, and Web*.
- Worboys, M. & Duckham, M. (2004). *GIS: A Computing Perspective*. CRC Press.
- Yeung, A. & Hall, G.B. (2007). *Spatial Database Systems: Design, Implementation, and Project Management*. Springer.

SOBRE LA AUTORA

Elzbieta Malinowski: Ingeniera en Computación, Doctora en Ciencias Aplicadas. Docente de la Escuela de Ciencias de la Computación e Informática. Teléfono 2511-8000. Correo electrónico: elzbieta.malinowski@eccu.ucr.ac.cr

