

Herramienta de visualización del proceso de optimización heurística de consultas SQL

Trabajo Terminal No. ____

*Alumnos: Guzmán Nieto Edwin David, *Salazar Pérez Yhoselin Guadalupe*

Directores: Chavarria-Báez Lorena, Ocotitla Rojas Nancy

**e-mail: ysalazarp1400@alumno.ipn.mx*

Resumen – A continuación, se plantea una herramienta con interfaz web que ayude a visualizar y comprender la optimización de consultas SQL por el método heurístico mediante la traducción de ésta a una expresión en Álgebra Relacional y la estructura del árbol canónico de manera que al observar todo este proceso los estudiantes y/o profesionales interesados en bases de datos se familiaricen con esta estrategia de optimización.

Palabras clave – Álgebra relacional, aplicación web, bases de datos, optimización heurística de consultas SQL.

1. Introducción

Las bases de datos son de vital importancia para los sistemas de información hoy en día pues la cantidad de datos que se manejan en ellos a diario es enorme y se debe asegurar su estabilidad, coherencia y permanencia. Surge la necesidad de que la manera en la que se guarda, procesa y se visualiza esta información sea cada vez mas eficiente. Gracias a las bases de datos relacionales, los desarrolladores pueden visualizar la organización de los datos con tablas que representan cómo están relacionados y así dar una respuesta más rápida ante consultas complejas en comparación con otros sistemas de bases de datos sin embargo, cuando se llega al área de lenguaje SQL (Structured Query Language), concretamente cuando se requiere obtener algún dato en específico, se dificulta encontrar un plan de consulta eficiente que consiga la información que se quiere obtener de la manera más óptima, esto es muy importante ya que existe un gran número de opciones que pueden generar el mismo resultado y para procesar cada consulta primero se analiza, es decir, se convierte en un árbol de análisis para representar la estructura de una manera más útil, después éste árbol es transformado en un plan de consulta lógico, por ejemplo, el árbol de expresión algebraica para finalmente convertirlo en un plan de consulta físico que indica, entre otros aspectos, las operaciones realizadas y su orden de ejecución. Es un proceso largo y que dependiendo de la estructura de la consulta puede complicarse o resolverse de una mejor manera. Si bien en la ejecución de una consulta se puede verificar si la respuesta que se obtiene es la correcta, hacen falta herramientas que permitan visualizar todo el proceso que se lleva a cabo en el plan lógico y optimización de una consulta SQL.

Aunque se cuenten con algunos sistemas gestores de bases de datos (SGBD) que ofrecen optimizar las consultas siguiendo diferentes estrategias, éstas no muestran el plan de ejecución en álgebra relacional y son exclusivas del propio sistema gestor, además, en estos casos, el desarrollador tiene que familiarizarse con la simbología y demás aspectos para comprender el plan de ejecución de consulta, esto para SQL Server por ejemplo. Varias instituciones académicas en otros países también han desarrollado herramientas para el apoyo a estudiantes en el área de consultas SQL y/o álgebra relacional como las citadas en la tabla 1, sin embargo, es complicado acceder a la herramienta si no se es parte de su comunidad además de tener diferentes especificaciones señaladas en la columna de características en la misma tabla. Hasta donde se ha investigado no se ha encontrado un trabajo terminal o tesis a nivel nacional que ofrezca una solución a esta problemática.

SOFTWARE	CARACTERÍSTICAS	PRECIO EN EL MERCADO
“ANACON: analizador de consultas SQL.” Universidad de Zaragoza – 2007 España	Herramienta de apoyo para las prácticas de la asignatura Bases de Datos I en la Escuela Universitaria Politécnica de Teruel. ANACON, es un analizador de consultas SQL que permite que los alumnos puedan aprender de forma más rápida e intuitiva a realizar consultas en dicho lenguaje, sin tener que conocer los aspectos más complejos y las particularidades que suelen presentar los analizadores de consultas que integran los sistemas gestores de bases de datos relacionales (SGBD) [1]. No cuenta con optimización de consultas, no implementa álgebra relacional.	Uso gratuito para la Escuela Universitaria Politécnica de Teruel

<p>“A Learning Tool Visualizing the Algebraic Laws for Optimizing Query Execution”</p> <p>International Conference on e-Learning and the Knowledge Society, 2010, Bulgaria</p>	<p>Herramienta de aprendizaje interactiva que visualiza la misma pregunta a una base de datos de diferentes formas usando expresiones equivalentes y manteniendo el mismo resultado. La herramienta muestra el costo de tiempo de la ejecución de expresiones equivalentes que permite a los estudiantes comparar y elegir la declaración SQL óptima y eficiente. Se dan las leyes algebraicas sobre operaciones relacionales y su implementación en el sistema. [2]</p>	<p>Uso gratuito en el curso de Bases de datos de la Universidad de Ruse en 2010</p>
<p>“Herramienta docente para consultas en Oracle”</p> <p>Universidad de Burgos, 2004, España</p>	<p>Herramienta docente que sirve de ayuda en la labor de aprendizaje del alumno en cuanto a Bases de Datos se refiere, y más concretamente que sirva de soporte a la asignatura de Administración de Bases de Datos, ayudando al alumno en el manejo y aprendizaje de consultas en Oracle, y en la profundización en las tareas propias del administrador de bases de datos, a través de la Web [3]. Con opción a procesar y optimizar una consulta de tres maneras: basada en costes, basada en reglas o mediante hints. Interfaz web de conexión exclusiva a bases de datos de Oracle.</p>	<p>Etapas de pruebas en la comunidad de la Universidad de Burgos en el año 2005</p>
<p>“RAT Relational Algebra Translator”</p> <p>Universidad nacional de Costa Rica, 2013, Costa Rica</p>	<p>Permite escribir expresiones en álgebra relacional que se traducen a lenguaje SQL, realiza conexiones a bases de datos relacionales entre las que se encuentran: Oracle, MySQL, SQL Server, Access. Visualización de árboles binarios para álgebra relacional.</p>	<p>Uso libre, descarga disponible a la fecha en www.sinfo.una.ac.cr (página web oficial de la UNA), última versión 2013 [4]</p>
<p>Solución Propuesta</p>	<p>Se plantea desarrollar una aplicación web para analizar y optimizar consultas SQL mediante álgebra relacional y la visualización del árbol canónico como el plan de consulta lógico, con el objetivo de que se tenga una herramienta como apoyo en la comprensión, comprobación y visualización de la optimización heurística de consultas SQL.</p>	<p>Uso gratuito dentro de la ESCOM o posteriormente en el IPN.</p>

Tabla 1. Resumen de productos similares.

2. Objetivo

Implementar una herramienta gráfica con interfaz web que permita visualizar el proceso de optimización heurística de una consulta SQL.

3. Justificación

Al trabajar con bases de datos relacionales muy grandes se vuelve necesario contar con consultas que minimicen el tiempo o los pasos que ejecuta el procesador de consultas:

1. Árbol de análisis que representa la estructura.
2. Árbol de expresión algebraica (plan de consulta lógico).
3. Orden de ejecución de las Operaciones (plan físico)

Si la consulta está mal formulada puede generar desde la obtención de información errónea (error a nivel sintáctico, mostrar sólo una porción de los datos objetivo, resultado completamente diferente al requerido, etc.), hasta la prolongación de procesos innecesarios, aunque el resultado sea correcto. A futuro esto se ve reflejado en gasto de recursos que pueden afectar el cumplimiento de requerimientos en los sistemas.

Esto en muchos casos recae en el conocimiento o práctica que el desarrollador y/o administrador de bases de datos tenga para diseñar la consulta pues existen diferentes opciones para obtener la misma información, pero no todas ellas son precisamente la mejor manera o el mejor camino y llegar a ella sin tener una idea o visualización en general del proceso de ejecución de la consulta resulta complicado. Aunado a eso se ha investigado acerca de la existencia de alguna herramienta que sirviera a este fin. En ese sentido, hasta el momento no se ha encontrado una aplicación o sistema que realice esta actividad, que se enfoque en la optimización heurística y al mismo tiempo permita traducir y visualizar este proceso en álgebra relacional, al menos no de manera comercial.

Existen algunas aplicaciones desarrolladas en Universidades que imparten asignaturas relacionadas con Bases de Datos, pero sólo una es accesible en línea y fue desarrollada en el año 2013 sin soporte para años siguientes. Es por ello que se propone este proyecto pues es una idea bastante interesante a desarrollar, además de tener la posibilidad de compartir esta herramienta para un uso libre o en todo caso, comercial. La idea principal es desarrollar una herramienta con formato de aplicación web para visualizar el proceso de optimización de consultas SQL, de manera que ayude a las personas interesadas en el desarrollo y/o administración de bases de datos relacionales a verificar u obtener la consulta óptima con una estrategia basada en álgebra relacional y reglas de equivalencia enfocadas a operadores.

Esta herramienta permitirá que, mediante el ingreso de una consulta de en SQL, se muestre la parte lógica con su equivalente en álgebra relacional, con un árbol canónico que será la base para mejorar esta expresión y traducirla nuevamente a SQL. A diferencia de la mayoría de las herramientas académicas, se plantea que se conecte a más de una opción de bases de datos para poder probar y verificar la consulta, pues solemos trabajar con diferentes sistemas gestores y enfocarnos en uno, limitaría el uso de la aplicación, además algunas de estas herramientas sólo ofrecen un visualizador de la consulta, parecido a los sistemas gestores de bases de datos en el que no se observa el plan de consulta y esto se vuelve de vital importancia para el entendimiento sobre la ejecución de consultas complejas sobre todo haría que el usuario se familiarice con esta transición entre SQL y álgebra relacional o viceversa. Este sería otro aporte pues, lo que se ha implementado hasta ahora, sólo permite traducir de SQL a expresión en álgebra relacional o viceversa pero no las dos opciones en la misma herramienta. Se plantea que se puedan ingresar consultas de aproximadamente 7 a 8 operadores en su equivalente en Álgebra Relacional.

Al tener una interfaz web a diferencia por ejemplo de ANACON y RAT (véase tabla 1) permitiría una mayor disponibilidad y facilidad de uso, así como un entorno más amigable para los que sólo quieren comprobar o mejorar sus planes de consulta. Pasando a la parte de Álgebra Relacional, no disponemos de un mecanismo que permita interactuar con las bases de datos de la actualidad directamente [2], por lo que este proyecto apoyaría a la comprensión tanto del proceso SQL como a su expresión equivalente lógica.

4. Productos o Resultados esperados

El proyecto se presentará como una interfaz web en la cual se incluirá principalmente un editor de consultas SQL (entrada) para que sea procesada por el intérprete y se traduzca a su expresión en Álgebra Relacional, de esta manera se podrá observar el plan de consulta lógico y será mediante éste que se obtendrá la optimización. El usuario visualizará el árbol canónico generado para comprender el paso a paso de todo el proceso, finalmente este resultado (plan de consulta optimizado) se traducirá nuevamente a sentencia SQL.

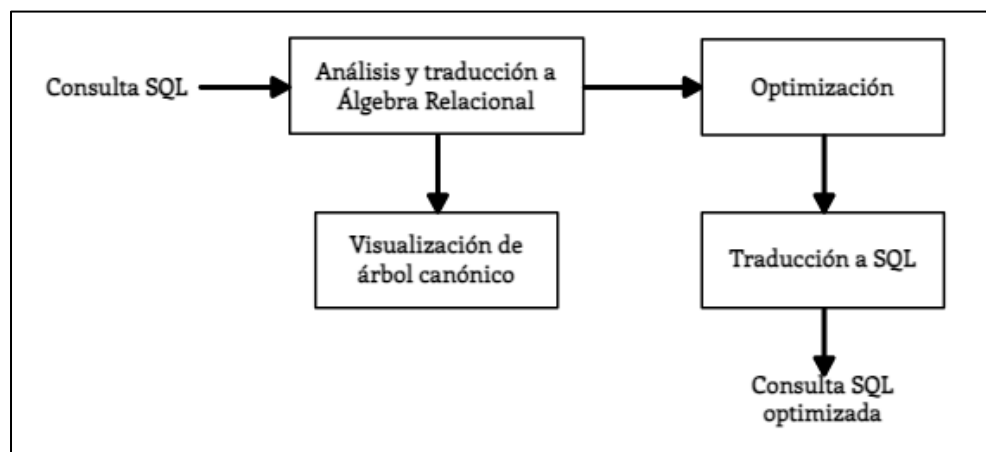


Figura 1. Arquitectura del sistema.

Los productos esperados de este trabajo terminal se muestran en forma de lista a continuación:

1. El código.
2. La documentación técnica del sistema.
3. El manual de usuario (se incluirá en la página principal)

5. Metodología

Para el caso de la metodología vamos a utilizar Scrum, la cual es una metodología ágil que tiene como principal objetivo regular un conjunto de buenas prácticas al momento de trabajar colaborativamente, en equipo, garantizando el mejor resultado posible dentro del desarrollo del proyecto [7]. Otro punto importante por el cual hemos escogido esta metodología es porque está diseñada para entornos complejos, en donde se necesitan obtener resultados pronto y donde los requisitos podrían cambiar, teniendo la capacidad de cumplir con todos estos requisitos y objetivos.

Se ejecutarán ciclos temporales cortos de 4 semanas, donde después de cada iteración se deberá tener un resultado completo. En ese sentido las actividades que se plantea llevar a cabo son las siguientes:

- **Planificación de la iteración.** El primer día de la iteración realizaremos la reunión de planificación de la iteración. Esta etapa contará con dos partes:
 1. **Selección de requisitos (2 horas).** En esta parte, nos reuniremos alumnos y directoras de Trabajo Terminal con el fin de establecer los requisitos, objetivos y priorizar las actividades a desarrollar. En ese sentido, cada integrante pregunta las dudas que le surgieron sobre el proyecto y se seleccionan los requisitos más prioritarios que con respecto al análisis, se prevén que se podrán completar en la iteración, de manera que se tenga la capacidad de entregarlos antes de terminar la siguiente iteración.
 2. **Planificación de la iteración (2 horas).** Elaboraremos la lista de tareas (cronograma) de la iteración necesarias para desarrollar los requisitos seleccionados. La estimación de esfuerzo se hace de manera conjunta y nos asignamos las tareas que cada quien va a cumplir, nos organizamos para trabajar en conjunto con el fin de compartir conocimiento (creando un equipo más resiliente) o para resolver juntos objetivos especialmente complejos.
- **Ejecución de la iteración.** Cada día realizaremos una reunión de sincronización (15 minutos), normalmente delante de un tablero (Scrum Taskboard) virtual mediante el uso de Trello, como tecnología. Vamos a inspeccionar el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder realizar las adaptaciones necesarias que nos permitan cumplir con la previsión de objetivos a mostrar al final de la iteración. En la reunión cada miembro del equipo se encarga de responder a tres preguntas:
 - ✓ ¿Qué he hecho desde la última reunión de sincronización para ayudar al equipo a cumplir su objetivo?
 - ✓ ¿Qué voy a hacer a partir de este momento para ayudar al equipo a cumplir su objetivo?
 - ✓ ¿Qué impedimentos tengo o voy a tener que nos impidan conseguir nuestro objetivo?

Durante la iteración, nos encargamos de replantear la lista de requisitos (para prepararlos para las siguientes iteraciones) y, si es necesario, se cambian o replanifican los objetivos del proyecto (10%-15% del tiempo de la iteración) con el objetivo de maximizar la utilidad de lo que se desarrolla y el retorno de inversión.

- **Inspección y adaptación.** El último día de la iteración realizaremos una reunión de revisión de la iteración. Esta etapa cuenta con dos partes:
 1. **Revisión (demostración) (1,5 horas).** Presentamos los requisitos completados en la iteración, en forma de incremento de la aplicación web para ser entregado con el mínimo esfuerzo. De acuerdo con los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, se pueden realizar las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, replanificando el proyecto.
 2. **Retrospectiva (1,5 horas).** Analizamos cómo ha sido nuestra manera de trabajar y cuáles son los problemas que nos impiden progresar adecuadamente, mejorando de manera continua nuestra productividad. Además, se van a eliminar o escalar los obstáculos identificados que estén más allá del ámbito de acción del equipo.

Por último, cabe mencionar que esta metodología se adecua bastante bien al desarrollo de nuestro proyecto, de hecho, se tiene un tablero dentro de Trello, como ya se ha mencionado anteriormente, en el cual podemos observar de manera gráfica los avances y la información que estamos trabajando, además de algunos tiempos establecidos para poder realizar las actividades que nos hemos asignado entre los integrantes de este equipo.

6. Cronograma

De acuerdo con la metodología seleccionada para el desarrollo de este proyecto, los cronogramas siguientes fueron divididos por semanas, esto debido a que los sprints o iteraciones, se miden mediante este rubro. Las actividades están diseñadas para que los integrantes trabajen en conjunto cada una de las etapas del ciclo [11] de vida del proyecto y se establezca en cada planeación (primera semana de cada mes) las actividades individuales. Los cronogramas se encuentran al final del documento.

7. Referencias

- [1] Garrido, P., Martínez, F., Tramullas, J. y Fuertes, G. (2007). ANACON: ANALIZADOR DE CONSULTAS SQL. RIED. *Revista Iberoamericana de Educación a Distancia.*;10(1):201-215. ISSN: 1138-2783. Disponible en: <https://www.redalyc.org/articulo.oa?id=331427206012>
- [2] Villalobos, J., Brenes, S., y Mora, S. (2012). Computer-aided tool for the teaching of relational algebra in data base courses. *Uniciencia*, 26(1-2), 179-195. Disponible en: <https://www.revistas.una.ac.cr/index.php/uniciencia/article/view/3873>
- [3] García M.y Renedo, E. (2004). Herramienta docente para consultas en Oracle. X Jornadas de Enseñanza Universitaria de la Informática: JENUI 2004. Alicante. págs. 161-168 - ISBN 84-9732-334-3. Disponible en: <https://dialnet.unirioja.es/servlet/articulo?codigo=6512905>
- [4] Li, C., Soliman, M. A., Chang, K. C. C., y Ilyas, I. F. (2005). RankSQL: supporting ranking queries in relational database management systems. In *Proceedings of the 31st international conference on Very large data bases (pp. 1342-1345)*. Disponible en: <https://www.vldb.org/conf/2005/papers/p1342-li.pdf>
- [5] Arsova, E., y Arsov, S. (2010). A Learning Tool Visualizing the Algebraic Laws for Optimizing Query Execution. In *International Conference on E-Learning and Knowledge Society (pp. 94-99)*. Disponible en: <https://web.archive.org/web/20180410021254id/http://elearning-conf.eu/docs/cp10/paper-12.pdf>
- [6] Chavarría-Báez, L., y Rojas, N. O. (2016). Sobre el uso de herramientas CASE para la enseñanza de bases de datos. *Sistemas, Cibernética e Informática*, 13(2), 51-56. Disponible en: <https://www.iiis.org/CDs2016/CD2016Spring/papers/CB321MR.pdf>
- [7] *Qué es SCRUM*, (2021). *Proyectos Ágiles*. Disponible en: <https://proyectosagiles.org/que-es-scrum/>
- [8] Kroenke, D. M. (2003). *Procesamiento de bases de datos: fundamentos, diseño e implementación*. México: Pearson Educación.
- [9] Rivero, E., Martínez, L., y Alonso, I. (2005). *Bases de datos relacionales: fundamentos y diseño lógico* (Vol. 20). España: Universidad Pontifica Comillas.
- [10] Silberschatz, A., Korth, H. F., Sudarshan, S., Pérez, F. S., Santiago, A. I., y Sánchez, A. V. (2002). *Fundamentos de bases de datos* (Vol. 11). Ciudad de México, México: McGraw-Hill.
- [11] García-Molina, H., Ullman, J. y Widom, J. (2009). *Chapter 16-The query compiler*. Database Systems. (pp 759-770). Pearson Prentice Hall ISBN 0-13-606701-8

8. Alumnos y directores

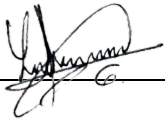
Guzmán Nieto Edwin David. - Alumno de la carrera de Ing. En Sistemas Computacionales en ESCOM, Especialidad Sistemas, Boleta: 2017630721, Tel. 5513117958, email: eguzmann1601@alumno.ipn.mx

Firma: _____



Salazar Pérez Yhoselin Guadalupe. - Alumna de la carrera de Ing. en Sistemas Computacionales plan 2009 en ESCOM, Especialidad Sistemas, Boleta: 2015630453, Tel. 5586797818, email ysalazarp1400@alumno.ipn.mx.

Firma: _____



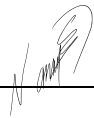
Lorena Chavarría Báez. - Dra. en Ciencias en Ingeniería Eléctrica opción Computación por el Centro de Investigación y de Estudios Avanzados del IPN (CINVESTAV-IPN), Profesora de la Escuela Superior de Cómputo, Áreas de Interés: Sistemas de información, bases de datos, email lorena_chavarria@yahoo.com.mx

Firma: _____



Ocotilla Rojas Nancy. - M. en C. en Computación del Centro de Investigación en Computación del IPN, Ing. en Sistemas Computacionales de ESCOM IPN, Profesora de ESCOM/IPN, Áreas de Interés: Bases de datos, Ingeniería de software y Desarrollo web, email nanwen1@gmail.com

Firma: _____



CARÁCTER: Confidencial
FUNDAMENTO LEGAL: Artículo 11 Fracc. V y Artículos 108, 113 y 117 de la Ley Federal de Transparencia y Acceso a la Información Pública.
PARTES CONFIDENCIALES: Número de boleta y teléfono.

CRONOGRAMA Nombre del alumno(a): Guzmán Nieto Edwin David

TT No.:

Título del TT: Herramienta de visualización del proceso de optimización heurística de consultas SQL

ACTIVIDAD	FEB	MAR	ABR	MAY	JUN	JUL	AGO	SEP	OCT	NOV	DIC
Planeación de lista de tareas o funcionalidades											
Análisis e investigación											
Diseño de sistema (por funciones)											
Generación de código											
Pruebas de cada sprint o iteración											
Maquetado y prueba de proceso: Visualización del árbol canónico.											
Evaluación de TT I.											
Revisión del proceso: Traducción a SQL											
Generación del Manual de Usuario en la Página web.											
Generación del Reporte Técnico.											
Evaluación de TT II											

CRONOGRAMA Nombre del alumno(a): Salazar Pérez Yhoselin Guadalupe

TT No.:

Título del TT: Herramienta de visualización del proceso de optimización heurística de consultas SQL

ACTIVIDAD	FEB	MAR	ABR	MAY	JUN	JUL	AGO	SEP	OCT	NOV	DIC
Planeación de lista de tareas o funcionalidades											
Análisis e investigación											
Diseño del sistema (por funciones)											
Generación de código											
Pruebas de cada sprint o iteración											
Maquetado y prueba de proceso: Análisis y traducción a Álgebra Relacional.											
Evaluación de TT I.											
Revisión del proceso de Optimización de la consulta											
Generación del Manual de Usuario en la Página web.											
Generación del Reporte Técnico.											
Evaluación de TT II											