



COMPARACIÓN DEL RENDIMIENTO DE LOS COMANDOS INSERT, SELECT Y DELETE EN LOS SISTEMAS GESTORES DE BASES DE DATOS ORACLE Y MYSQL¹

COMPARING THE EFFICIENCY OF THE INSERT, SELECT AND DELETE COMMANDS IN ORACLE AND MYSQL RELATIONAL DATABASE MANAGEMENT SYSTEMS
COMPARAÇÃO DO RENDIMENTO DOS COMANDOS INSERT, SELECT E DELETE NOS SISTEMAS GESTORES DE BANCOS DE DADOS ORACLE E MYSQL
Johan Alexander Portilla Tovar²; Mireya Bernal Gómez³

Fecha de recibido: Agosto 06 de 2017 | Fecha de aprobado: Octubre 03 de 2017

Resumen

Oracle y *Mysql* son gestores de bases de datos relacionales (con versiones libres) utilizados en el campo informático. Este artículo presenta los resultados de la comparación del rendimiento de los comandos *Insert*, *Select* y *Delete* de estos dos motores de bases de datos. Si bien la sintaxis de codificación de uno y otro gestor es similar, se tiende a pensar erróneamente que ocurre lo mismo con los resultados de ejecución de las sentencias mencionadas. Por esto, con este documento se demuestra cuál de estos dos motores de bases de datos es más eficiente al ejecutar cada uno de los comandos mencionados.

Se implementó un esquema de base de datos y se incluyeron varios registros con el comando *Insert*, para luego llevar a cabo los procesos con los comandos *Select* y *Delete*. Los resultados obtenidos se presentan de forma gráfica, en cuadros comparativos que evidencian qué motor de bases de datos es mejor para cada comando ejecutado con cierta cantidad de información, o, en su defecto, si el resultado obtenido en ambos motores de bases de datos es el mismo.

Palabras clave: Oracle, Mysql, SQL, DML DML, Gestores de Bases de datos relacionales.

1 Artículo de investigación.

2 Estudiante de Tecnología en Informática. Corporación Universitaria Minuto de Dios –UNIMINUTO. Correo electrónico: jportillato@uniminuto.edu.co

3 Docente de Tecnología en Informática. Corporación Universitaria Minuto de Dios –UNIMINUTO. Correo electrónico: mbernal@uniminuto.edu.co

Abstract

Oracle and MySQL are open-source relational database management systems used in computing. This article presents the results of a comparison of the efficiency of the Insert, Select and Delete commands when using these two database engines. Although both systems have a similar encoding syntax, people mistakenly think that the same is true for the execution results of the statements. Thus, through this paper we attempt to prove which of them is more efficient when executing the three commands mentioned.

We implemented a database schema and included a number of registers using the Insert command. Then, we did the same with the Select and Delete commands. The results are displayed graphically, in comparative charts showing whether one database engine is better for each command executed with a certain amount of information, or, otherwise, if the results obtained by using both database engines are the same.

Keywords: Oracle, MySQL, SQL, DML, relational database management systems

Resumo

Oracle e Mysql são gestores de bancos de dados relacionais (com versões livres) utilizados no campo informático. Este artigo apresenta os resultados da comparação do rendimento dos comandos Insert, Select e Delete destes dois motores de bancos de dados. Embora a sintaxe de codificação de um e outro gestor são semelhantes, se tende a pensar de maneira errónea, que ocorre o mesmo com os resultados de execução das sentenças mencionadas. Por isso, com este documento se demonstra qual destes dois motores de bancos de dados é mais eficiente ao executar a cada um dos comandos mencionados.

Implementou-se um esquema de banco de dados e incluíram-se vários registros com o comando Insert, para depois levar a cabo os processos com os comandos Select e Delete. Os resultados obtidos apresentam-se de forma gráfica, em quadros comparativos que evidenciam qué motor de bases de dados é melhor para cada comando executado com certa quantidade de informação, ou, em seu defeito, se o resultado obtido em ambos motores de bancos de dados é o mesmo.

Palavras-chave: Oracle, Mysql, SQL, DML, Relational Database Managers.

Esta investigación establece una comparación entre el rendimiento de dos motores de bases de datos relacionales (Oracle y Mysql), teniendo en cuenta los comandos *Insert*, *Select* y *Delete*, con el fin de dar a conocer por medio de pruebas computacionales cuál de estos sistemas gestores de bases de datos (SGBDR) es más eficiente a la hora de ejecutar los comandos mencionados, o, en su defecto, determinar si hay igualdad en las respuestas obtenidas. Para ello se realizó un recorrido por diferentes sentencias en Lenguaje Estructurado de Consultas (SQL) que fueron ejecutadas en ambos gestores de bases de datos, para puntualizar qué tan favorable puede ser cada uno; además, se muestran tablas de comparación con los resultados obtenidos en cada proceso de ejecución.

Se escoge a Oracle y Mysql como motores de bases de datos para realizar las pruebas, ya que permiten probar, con cierta cantidad de registros, el tiempo de respuesta que tardan en llevar a cabo la petición realizada (Cardona, Rendón & Reyes, 2016; García, 2017).

El enfoque que se da a este artículo está dirigido a un entorno educativo, ya que las plataformas y gestores con los que se realizan las pruebas son las que actualmente se usan en este ámbito. Además, puede ser de utilidad para quienes se inicien como desarrolladores en bases de datos, bien en un programa tecnológico, bien en uno profesional o en el área de la informática.

Según Medrano y Torreño (2016), entre los gestores de bases de datos Oracle y Mysql, en lo que respecta a su rendimiento, Mysql tiene mejores resultados en las sentencias de actualización y borrado de datos; pero Oracle es más rápido en la realización de consultas. De acuerdo con Zhingri y Augusto (2016), quienes comparan el rendimiento de Mysql con MongoDB, se determina que, en cuanto a seguridad, Mysql es la mejor opción; sin embargo, MongoDB es más veloz. En el estudio de Pilco Guachi (2012) se hace una comparación entre Mysql y Firebird, en el cual se evidencia que Firebird tiene un mejor desempeño en tiempos de respuesta en las consultas, aunque no sea superior en características. En este estudio se ejecutan tres comandos *Select*, con pocos registros, lo cual no genera sensación de una visualización de resultados completa.

La investigación de Inzunza (2016) realizó un análisis comparativo de tres gestores de bases de datos relacionales (SQL Server 2012, Postgre SQL y Mysql),

en donde se determina que Mysql y SQL Server 2012, al tener aspectos cualitativos iguales, presentaron comportamientos similares a la hora de dar respuestas a las consultas realizadas; sin embargo, Postgres SQL presentó mejores resultados sobre estos gestores.

Cuervo (2012), comparando el rendimiento de dos gestores de bases de datos relacionales, Mysql y Firebird, obtuvo como resultado que Mysql es más rápido al ejecutar las transacciones realizadas. No obstante, este artículo aclara que el criterio de resultado de todos los estudios que se hacen depende del proyecto en el que vayan a ser ejecutados, y, por lo tanto, no se puede decir si un motor de bases de datos es mejor que otro.

METODOLOGÍA

En esta investigación se llevaron a cabo los procedimientos relacionados a continuación:

1. Se instalaron dos motores de bases de datos relacionales: Mysql y Oracle. (En la Tabla 1 se presentan las características del equipo en el que se hizo la instalación).
2. Se ejecutaron las sentencias *Select*, *Insert* y *Delete* con una cantidad de datos que van desde 10 000 hasta 50 000 registros, con los cuales se lleva a cabo el proceso de comparación.
3. Se tabularon los resultados obtenidos en las comparaciones realizadas y se dedujeron las conclusiones.

Tabla 1. Computador utilizado

CARACTERÍSTICAS COMPUTADOR UTILIZADO PARA REALIZAR PRUEBAS	
Procesador	AMD Athlon II X4 645 HD Graphics 3.1 GHz
Memoria RAM	4,00 GB DDR3
Disco Duro	HTS SATA Disk Device 1 TB
Sistema Operativo	Windows 8 64 Bits
Video	AMD Radeon HD 6310 Graphics

Fuente: elaboración propia.

A continuación, se detallan los procedimientos realizados en la investigación:

Instalación y manipulación

Oracle: La herramienta usada es Oracle SQL Developer (versión libre), el manejo de base de datos con el programa ORACLE XE

Mysql: Se utiliza la plataforma de uso libre llamada XAMPP.

A continuación, en la tabla 2, se evidencian las características de los gestores de bases de datos utilizados.

Tabla 2. Características de los gestores usados

	Oracle	Mysql
Versión	11g	5.6.16
Plataforma de manejo	Oracle XE v11.2	XAMPP v3.2.1
Codificación	Oracle SQL Developer v4.2	PHPMyAdmin v4.7.1
Servidor	127.0.0.1:80 80	127.0.0.1 via TCP/IP

Fuente: elaboración propia.

El almacenamiento de cada gestor se muestra a continuación:

Mysql: En el disco duro y memoria RAM (no se requiere una capacidad específica) el computador nativo casero es suficiente, en este caso de 1 TB y 4 GB, respectivamente (Mysql, 2017).

Oracle: Solo permite 4 GB de almacenamiento y/o 1 GB de memoria RAM. Al superar este límite, Oracle ofrece otras versiones completas, en las que hay que pagar licencias (Oracle, 2017).

Al usar las versiones libres y no completas de estos dos gestores, lo que implica ciertas limitaciones, se estipula que las pruebas que se hacen aquí son enfocadas a educadores y estudiantes, ya que estos gestores se usan habitualmente para propósitos de enseñanza y aprendizaje. También está enfocado a quienes estén en introducción al campo de las bases de datos.

El proceso para la obtención de datos que se van a evaluar en las tablas es complejo. En este caso, se utiliza un *Script* de datos de uso libre, cuya autoría es anónima, pero que trae Oracle XE a la hora de su instalación en el usuario predefinido "HR". Este script contiene la creación de las tablas, sus

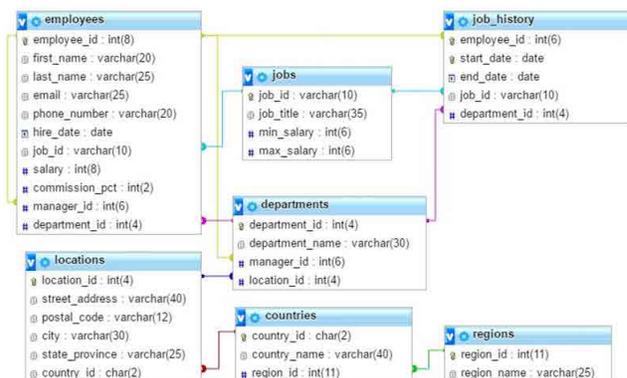
relaciones y la inserción de los datos necesarios, lo cual se puede descargar de:

Mysql: <http://bit.ly/2AxAo5g>

Oracle: <http://bit.ly/2hiiP0o>

Los *Scripts* son de uso libre y contienen las sentencias necesarias que se aplican en las pruebas. En este caso se usan con fines netamente didácticos, y se preserva la protección de datos personales, si así es el caso. Para estas pruebas, en el caso de Oracle XE, no se utiliza el usuario por defecto que trae "SYSTEM" ni tampoco el usuario cuyo esquema es "HR"; más bien, se crea un usuario y se copia el *Script* en él, para manejar la base de datos de manera autónoma. En el caso de Mysql no se realizan creaciones de usuario, simplemente se copia el *Script* para crear una base de datos. Al finalizar el proceso de la codificación de los *Scripts* en sus respectivos gestores, se obtiene el modelo Entidad-Relación de la base de datos, el cual se presenta en la Figura 1:

Figura 1. Modelo Entidad-Relación de la base de datos utilizada



Fuente: elaboración propia.

La carga de datos es similar en ambos gestores; sin embargo, hay ciertas diferencias en su codificación, y por eso cada sentencia que se evalúa aportará, bien para los dos gestores, o para cada uno por aparte, según sus diferencias. En estas sentencias se maneja el lenguaje SQL, el cual es el estándar en los Sistemas de Gestión de Bases de Datos (Torres, López & Flores, 2016, p. 1). Se cuenta con la ventaja de que los mismos gestores arrojan los tiempos verídicos de ejecución justo después de finalizar cada proceso.

A continuación, se realizará la ejecución de las sentencias a evaluar; se muestra una tabla de resultados obtenidos, un gráfico comparativo y un análisis del proceso.

La visión global de los comandos a valorar se presenta a continuación (Díez del Valle & Torreño, 2016, p. 47):

- Inserción de datos
- Consultas sencillas
- Consultas complejas
- Ordenaciones
- Agrupaciones
- Combinaciones entre tablas (Joins) simples
- Combinaciones entre tablas (Joins) múltiples
- Subconsultas simples
- Subconsultas múltiples
- Borrados

Pruebas Insert

Tipo: Inserción de datos

Descripción: Este comando engloba la creación de los registros necesarios en la base de datos. Se insertan los registros en la tabla `EMPLOYEES` de manera lógica, teniendo en cuenta sus relaciones en la base de datos.

El en siguiente vínculo puede descargarse del archivo que contiene el Script por ejecutar: <http://bit.ly/2q60sTg>

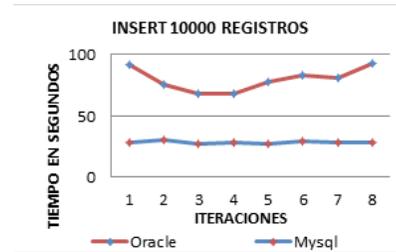
Para las pruebas, se insertaron 10 000, 20 000, 30 000, 40 000 y/o 50 000 registros, según corresponda. En la Tabla 3 se muestra la cantidad de registros insertados, los cuales van desde 10 000 hasta 50 000, determinando el tiempo de respuesta arrojado por los dos motores de bases de datos en cada una de las iteraciones realizadas (para este caso 8).

Tabla 3. Resultados prueba Insert

Registros→	10000		20000		30000		40000		50000	
	Tiempo (seg)		Tiempo (seg)		Tiempo (seg)		Tiempo (seg)		Tiempo (seg)	
Iteración→	Oracle	Mysql	Oracle	Mysql	Oracle	Mysql	Oracle	Mysql	Oracle	Mysql
1→	91,821	28,550	181,453	58,930	275,292	86,420	340,557	130,320	402,065	159,570
2→	75,307	30,980	200,088	72,090	279,462	82,590	371,165	123,270	419,304	165,700
3→	68,765	27,260	175,143	62,630	256,088	79,010	355,082	137,240	439,653	160,340
4→	68,606	28,250	201,139	70,450	270,024	83,330	373,458	137,196	427,174	158,070
5→	77,859	27,090	200,932	60,270	268,992	85,090	338,670	131,656	411,553	146,590
6→	83,789	29,020	191,609	59,230	282,342	80,010	349,099	129,116	398,799	143,150
7→	81,232	28,580	204,600	54,900	270,077	88,150	365,032	140,577	423,226	142,290
8→	92,614	28,220	179,091	59,210	262,552	78,300	354,546	138,465	404,079	149,644
Promedio→	79,999	28,494	191,757	62,214	270,604	82,863	355,951	133,480	415,732	153,169

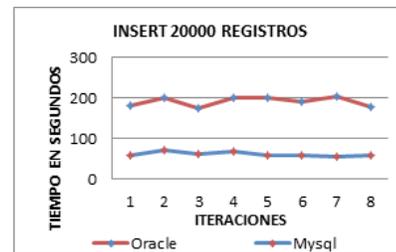
Fuente: elaboración propia.

Figura 2. Resultados prueba Insert 10 000
Resultados obtenidos por los dos motores de bases de datos al insertar 10 000 registros en 8 iteraciones



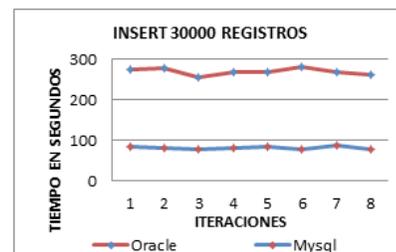
Fuente: elaboración propia.

Figura 3. Resultados prueba Insert 20 000
Resultados obtenidos por los dos motores de bases de datos al insertar 20 000 registros en 8 iteraciones



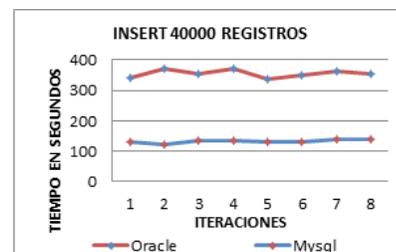
Fuente: elaboración propia.

Figura 4. Resultados prueba Insert 30 000



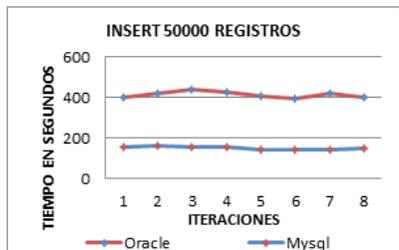
Fuente: elaboración propia.

Figura 5. Resultados prueba Insert 40 000
Resultados obtenidos por los dos motores de bases de datos al insertar 40 000 registros en 8 iteraciones



Fuente: elaboración propia.

Figura 6. Resultados prueba Insert 50 000
Resultados obtenidos por los dos motores de bases de datos al insertar 50 000 registros en 8 iteraciones



Fuente: elaboración propia.

Al analizar las figuras 2, 3, 4, 5 y 6 se observa que el tiempo de ejecución de la sentencia *Insert* es demasiado alto en ambos gestores; sin embargo, Mysql logra una mayor eficiencia, ya que obtiene un tiempo casi 3 veces menor. Mysql demoró en promedio 92 segundos, mientras que Oracle unos 262 segundos. Se observa de igual manera que Mysql oscila entre los 27 y 165 segundos al ejecutar la inserción de los datos, mostrando así constancia en cada proceso; en cambio Oracle muestra mucha variación, oscilando entre 68 y 439 segundos. 10 000 registros insertados en Mysql demoran aproximadamente 31 segundos; mientras que, en Oracle, unos 84 segundos.

Pruebas *Select*

En la Tabla 4 se enlistan los comandos de consulta que se utilizaron para evaluar el rendimiento en ambos motores de bases de datos. De cada sentencia se realizaron 5 iteraciones, con el fin de tomar los tiempos de ejecución y sacar el tiempo promedio (como se muestran en las Tablas 5, 6, 7, 8, 9, 10, 11, 12, 13; y luego en las Figuras 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21 se muestra el análisis gráfico que describe de los resultados obtenidos.

Tabla 4. Sentencias *Select* a evaluar

Grupo No.	Tipo de Agrupación	Consulta(s)
1	Consulta sencilla	1, 2
2	Consulta única	3
3	Condición múltiple	4,5
4	Ordenación	6
5	Agrupación y condiciones de grupo	7,8
6	Combinación entre tablas (JOIN SIMPLE)	9
7	Combinación entre tablas (JOIN MÚLTIPLE)	10
8	Subconsulta simple	11
9	Subconsulta múltiple	12

Fuente: Tipos de consulta que se llevan a cabo (Díez del Valle & Torreño, 2016).

Consulta número 1

Tipo: Consulta sencilla

Descripción: Traer todos los registros de la tabla EMPLOYEES.

Sentencia para los dos gestores: *Select * from Employees.*

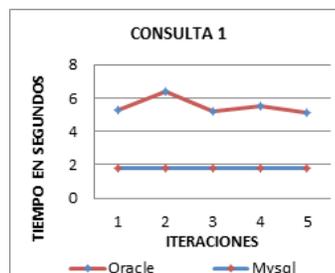
Datos recuperados: 50107

Tabla 5. Resultados de la consulta número 1 en los dos motores de bases de datos

Iteración↓	Tiempo (seg)	
	Oracle	Mysql
1→	5,583	1,804
2→	6,571	1,815
3→	5,385	1,808
4→	5,496	1,808
5→	5,356	1,800
Promedio	5,678	1,807

Fuente: elaboración propia.

Figura 7. Resultados de la consulta número 1 obtenidos por los dos motores de bases de datos



Fuente: elaboración propia.

De acuerdo con la Tabla 5 y la Figura 7, se puede decir que, al ejecutar una sentencia de consulta sencilla, el tiempo de ejecución en Mysql es 3 veces inferior, en promedio, que en Oracle, ya que varía en 1,2 segundos, de acuerdo a las 5 iteraciones realizadas, mientras que Mysql varía en solo 0,01 segundos, mostrando así mayor efectividad en la ejecución de la respuesta.

Consulta número 2

Tipo: Consulta sencilla

Descripción: Traer el nombre y cargo concatenado de la tabla EMPLOYEES.

Sentencia para Oracle: *Select first_name || 'Es un' || job_id as "cargos" from employees.*

Sentencia para Mysql: *Select CONCAT (first_name, 'Es un', job_id) as "cargos" from employees*

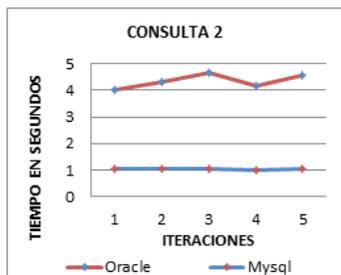
Datos recuperados: 50107

Tabla 6. Promedio de resultados de la consulta número 2 en los dos motores de base de datos

Iteración↓	Tiempo (seg)	
	Oracle	Mysql
1→	4,000	1,032
2→	4,304	1,026
3→	4,653	1,032
4→	4,174	1,024
5→	4,553	1,038
Promedio	4,337	1,030

Fuente: elaboración propia.

Figura 8. Resultados de la consulta número 2 obtenidos por los dos motores de bases de datos



Fuente: elaboración propia.

De acuerdo con la Tabla 6 y la Figura 8 se puede decir que, al ejecutar una sentencia de consulta concatenada sencilla, en Mysql el tiempo de respuesta es 4 veces más rápido, en promedio, que en Oracle. Además, el tiempo varía más en Oracle que en Mysql.

Consulta número 3

Tipo: Consulta única

Descripción: Traer los campos ID y cargo de la tabla EMPLOYEES cuyo id sea "10200"

Sentencia para los dos gestores: *Select employee_id, job_id from employees where employee_id = '10200'*

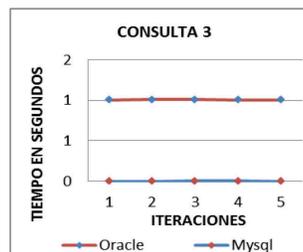
Datos recuperados: 1

Tabla 7. Resultados de la consulta número 3 obtenidos por los dos motores de bases de datos

Iteración↓	Tiempo (seg)	
	Oracle	Mysql
1→	0,892	0,0012
2→	0,900	0,0013
3→	0,898	0,0018
4→	0,896	0,0017
5→	0,894	0,0016
Promedio	0,896	0,0015

Fuente: elaboración propia.

Figura 9. Resultados de la consulta número 3 obtenidos por los dos motores de bases de datos



Fuente: elaboración propia.

De acuerdo con la Tabla 7 y Figura 9, se puede decir que, al ejecutar una sentencia de consulta única, los tiempos promedios medidos por ambos gestores se mantienen constantes: Mysql lo hace casi 1 segundo más rápido que Oracle.

Consulta número 4

Tipo: Condición múltiple

Descripción: Traer los campos nombre, apellido y email de la tabla EMPLOYEES que comiencen con las letras B, C, D, E, F, G o H, mediante el operador OR.

Sentencia para los dos gestores: *select first_name, last_name, email from employees where first_name like 'B%' or first_name like 'C%' or first_name like 'D%' or first_name like 'E%' or first_name like 'F%' or first_name like 'G%' or first_name like 'H%'.*

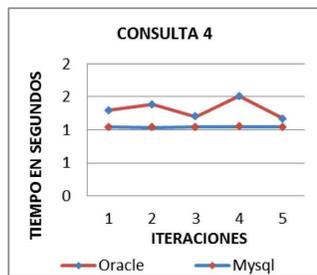
Datos recuperados: 18561

Tabla 8. Resultados de la consulta número 4 en los dos motores de bases de datos

Iteración↓	Tiempo (seg)	
	Oracle	Mysql
1→	1,295	1,040
2→	1,383	1,036
3→	1,197	1,042
4→	1,508	1,047
5→	1,168	1,041
Promedio	1,310	1,041

Fuente: elaboración propia.

Figura 10. Resultados obtenidos de la consulta número 4 por los dos motores de bases de datos



Fuente: elaboración propia.

De acuerdo con la tabla 8 y la Figura 10, que comparan los tiempos de ejecución de una condición múltiple, se puede decir que Oracle presenta una mejor respuesta frente a las consultas anteriores; sin embargo, Mysql logra sobreponerse por un promedio de 0,3 segundos.

Consulta número 5

Tipo: Condición múltiple

Descripción: Traer los campos nombre y el teléfono de los registros de la tabla EMPLOYEES cuyo departamento sea 50,60, 80, 100 mediante el operador IN.

Sentencia para los dos gestores: *Select first_name, phone_number from employees where department_id IN (50, 60, 80, 100);*

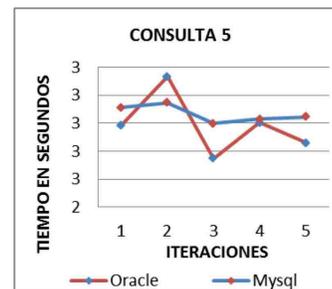
Datos recuperados: 47090

Tabla 9. Resultados de la consulta número 5 en los dos motores de bases de datos

Iteración↓	Tiempo (seg)	
	Oracle	Mysql
1→	2,595	2,627
2→	2,683	2,636
3→	2,537	2,599
4→	2,601	2,607
5→	2,565	2,611
Promedio	2,596	2,616

Fuente: elaboración propia.

Figura 11. Resultados obtenidos de la consulta número 5 por los dos motores de bases de datos



Fuente: elaboración propia.

De acuerdo con la Tabla 9 y la Figura 11, se puede observar que, al ejecutar una sentencia de consulta múltiple, Oracle gana en tiempo promedio por solo 0,02 segundos. Sin embargo, ambos gestores manifiestan mucha variación en sus tiempos de respuesta.

Consulta número 6

Tipo: Ordenación

Descripción: Traer los registros de la tabla EMPLOYEES cuyo salario esté entre 6000 y 12 000, ordenado por la columna salario. Se utiliza el operador BETWEEN.

Sentencia para los dos gestores: *Select * from employees where salary BETWEEN 6000 and 12000 order by salary.*

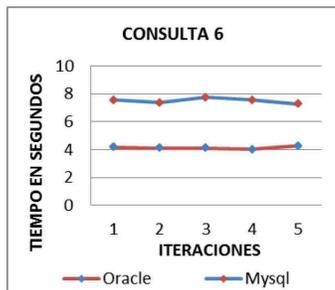
Datos recuperados: 23335

Tabla 10. Resultados de la consulta número 6 en los dos motores de bases de datos

Iteración↓	Tiempo (seg)	
	Oracle	Mysql
1→	4,173	7,555
2→	4,105	7,370
3→	4,115	7,725
4→	4,007	7,536
5→	4,252	7,255
Promedio	4,130	7,488

Fuente: elaboración propia.

Figura 12. Resultados obtenidos de la consulta número 6 por los dos motores de bases de datos



Fuente: elaboración propia.

De acuerdo con la Tabla 10 y la Figura 12, se observa que, al ejecutar una sentencia de ordenación de datos, Oracle tiene un tiempo de respuesta más prolongado que el de Mysql, el cual lo hace casi el doble de rápido. Su diferencia total promedio es de 3,3 segundos. Se muestra constancia en ambos gestores.

Sentencias *Delete*

A continuación, en la Tabla 11, se muestra la cantidad de registros, que van de 10 000 a 50 000, con respecto a los cuales se determina el tiempo de respuesta de los dos gestores frente a la sentencia *Delete*. El proceso se lleva a cabo con un total de 8 iteraciones, para una mayor efectividad del estudio.

Tipo: Borrado

Descripción: Borrar los registros de los empleados.

Sentencia para los dos gestores: *Delete from employees where employee_id between 207 and **¹

Datos borrados: 10 000*, 20 000*, 30 000*, 40 000* y/o 50 000*.

Tabla 11. Resultados de la prueba Delete

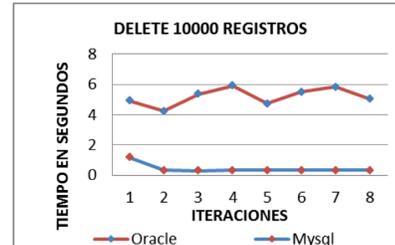
Registros	10000		20000		30000		40000		50000	
	Tiempo (seg)		Tiempo (seg)		Tiempo (seg)		Tiempo (seg)		Tiempo (seg)	
Iteración↓	Oracle	Mysql								
1→	4,932	1,184	6,121	0,637	8,003	0,870	10,726	0,988	12,965	1,240
2→	4,246	0,333	6,598	0,653	9,607	0,905	10,442	0,995	13,341	1,312
3→	5,355	0,305	7,355	0,644	8,432	0,853	11,002	1,003	13,347	1,201
4→	4,724	0,313	6,504	0,656	9,235	0,863	10,126	1,076	13,276	1,200
5→	4,724	0,334	6,209	0,636	8,234	0,899	10,234	0,983	12,812	1,106
6→	5,507	0,326	6,890	0,650	8,279	0,859	10,548	1,009	12,775	1,258
7→	5,823	0,328	5,289	0,624	9,926	0,901	10,962	1,028	12,491	1,157
8→	5,021	0,322	6,095	0,635	9,301	0,852	10,803	1,004	13,035	1,250
Promedio	5,439	0,431	6,383	0,642	8,877	0,875	10,605	1,011	13,005	1,216

Fuente: Tiempo de respuesta en borrado de datos con cierto número de registros (Almonacid, 2016).

A continuación, se muestra en las Figuras 13, 14, 15, 16 y 17 el tiempo de ejecución de borrado de acuerdo a la cantidad de registros.

Figura 13. Resultados prueba Delete 10 000

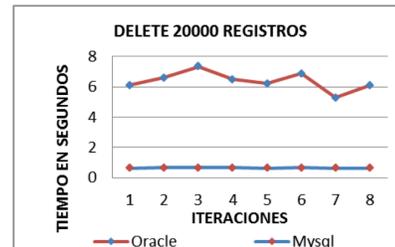
Resultados obtenidos en tiempo por los dos motores de bases de datos al llevar a cabo el proceso de borrar 10 000 registros



Fuente: elaboración propia.

Figura 14. Resultados prueba Delete 20 000

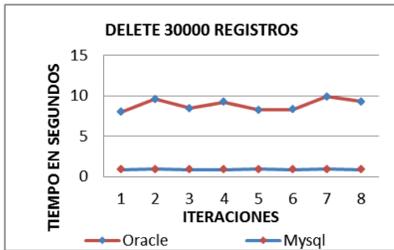
Resultados obtenidos en tiempo por los dos motores de bases de datos al llevar a cabo el proceso de borrar 20 000 registros



Fuente: elaboración propia.

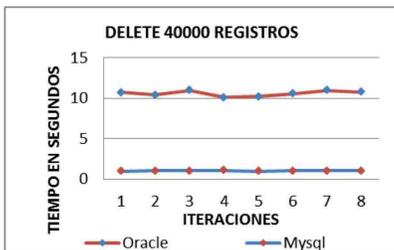
1 El asterisco corresponde al dato donde va a llegar.

Figura 15: Resultados prueba Delete 30 000
Resultados obtenidos en tiempo por los dos motores de bases de datos al llevar a cabo el proceso de borrar 30 000 registros



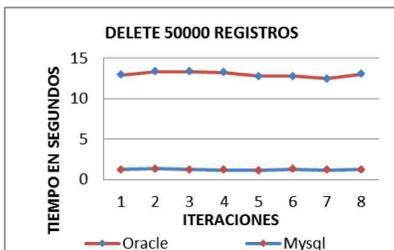
Fuente: elaboración propia.

Figura 16. Resultados prueba Delete 40 000
Resultados obtenidos en tiempo por los dos motores de bases de datos al llevar a cabo el proceso de borrar 40 000 registros



Fuente: elaboración propia.

Figura 17. Resultados prueba Delete 50000
Resultados obtenidos en tiempo por los dos motores de bases de datos al llevar a cabo el proceso de borrar 50 000 registros



Fuente: elaboración propia.

De acuerdo con las Figuras 13, 14, 15, 16, y 17, se observa que el tiempo de ejecución de la sentencia DELETE es muy extenso en el gestor Oracle, con un promedio total de 8,8 segundos en todo el proceso. En cambio, el gestor Mysql logra una mayor eficiencia, con un promedio total de 0,8 segundos en todo el proceso, es decir, que logra ejecutar la sentencia 9 veces más rápido.

Se observa asimismo que el gestor Mysql oscila entre los 0,3 y 1,3 segundos al ejecutar la inserción de los datos, mostrando así constancia en cada proceso. Por el contrario, Oracle muestra mucha variación, ya que sus tiempos de respuesta oscilan entre 4,2 y 13,3 segundos. Por cada 10000 registros borrados en el gestor Mysql, este se demora aproximadamente 0,19 segundos.

RESULTADOS Y DISCUSIÓN

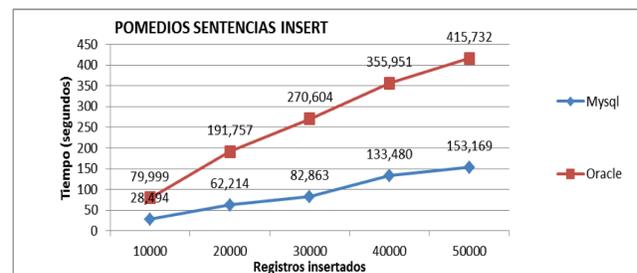
A partir de las pruebas efectuadas y los resultados obtenidos (indicados en la Figura 18), se puede determinar que el sistema gestor Mysql obtuvo un mejor rendimiento en las sentencias de inserción (INSERT) y borrado (DELETE) de registros, demostrando más rapidez frente al gestor Oracle. Mysql es 1.8 veces, en promedio, más eficiente que Oracle. Los datos se presentan a continuación en las Figuras 18 y 19:

Tabla 12. Resultados finales pruebas Delete e Insert

Sentencia→	Borrado		Inserción		Mejor Gestor
	Tiempo (seg)		Tiempo (seg)		
Registros Estudiados	Promedio Oracle	Promedio Mysql	Promedio Oracle	Promedio Mysql	
10000	5,439	0,431	79,999	28,494	Mysql
20000	6,383	0,642	191,757	62,214	Mysql
30000	8,877	0,875	270,604	82,863	Mysql
40000	10,605	1,011	355,951	133,480	Mysql
50000	13,005	1,216	415,732	153,169	Mysql
PROMEDIO TOTAL→	8,8618	0,835	262,808475	92,044	Mysql

Fuente: elaboración propia.

Figura 18. Gráfico comparativo final sentencias Insert



Fuente: elaboración propia.

Figura 19. Gráfico comparativo final sentencias Delete



Fuente: elaboración propia.

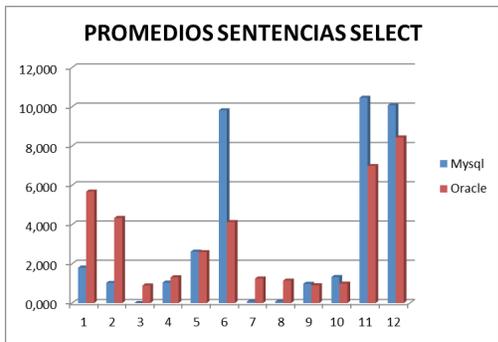
De acuerdo con el análisis, se determina que en el contexto de las sentencias de consultas (SELECT), hay mayor eficiencia en Oracle con respecto a Mysql, como se muestra en la Tabla 13 y la Figura 20:

Tabla 13. Resultados finales sentencias Select

Grupo No.	Tipo de Agrupación	Consulta(s)	Registros Obtenidos	Tiempo (seg)		Mejor Gestor
				Promedios Oracle	Promedios Mysql	
1	Consulta sencilla	1	50107	5,678	1,807	Mysql
		2	50107	4,337	1,030	Mysql
2	Consulta única	3	49106	0,896	0,002	Mysql
3	Condición múltiple	4	18561	1,310	1,041	Mysql
		5	47090	2,596	2,616	Mysql
4	Ordenación	6	23335	4,130	9,821	Oracle
5	Agrupaciones y condiciones de grupo	7	19	1,249	0,082	Mysql
		8	9	1,146	0,071	Mysql
6	Combinación entre tablas (JOIN SIMPLE)	9	21053	0,907	0,982	Oracle
7	Combinación entre tablas (JOIN MÚLTIPLE)	10	8018	0,991	1,328	Oracle
8	Subconsulta simple	11	13842	6,982	10,463	Oracle
9	Subconsulta múltiple	12	8781	8,439	10,077	Oracle
PROMEDIO TOTAL →				3,222	3,277	Oracle

Fuente: Consultas realizadas y resultados obtenidos en los dos motores de bases de datos con la sentencia Select (Diez del Valle & Torreño, 2016).

Figura 20. Gráfico comparativo final sentencias select



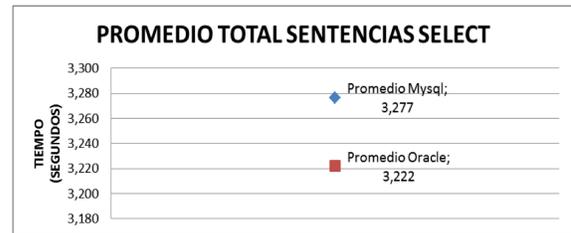
Fuente: elaboración propia.

Con el análisis realizado se determina que Oracle, a pesar de no haber demostrado buen funcionamiento

en la mayoría de sentencias Select, logra superar a Mysql en el total de los resultados. Es de notar también que Oracle realiza más rápido las sentencias de consultas compuestas, mientras que Mysql lo hace solo en las sentencias de consultas básicas.

A continuación, en la Figura 21 se gráfica el rendimiento de Oracle, al cual le tomó menos tiempo, en promedio, que a Mysql para realizar las consultas. Sin embargo, su ventaja es solo de 0,05 segundos.

Figura 21. Gráfico comparativo promedio total sentencias Select



Fuente: Resultados finales del promedio de los dos motores de bases de datos aplicando las diferentes sentencias Select (Inzunza, 2016).

De acuerdo con la investigación realizada, se puede decir que el rendimiento en ambos motores de bases de datos es similar a la hora de realizar sentencias de consultas, pero no a la hora de insertar o borrar datos, ya que en este caso Mysql tiene ventaja frente a Oracle.

CONCLUSIONES

Este estudio ha logrado una revisión exhaustiva del rendimiento de dos Sistemas de Gestión de Bases de Datos Relacionales, Mysql y Oracle XE, que son ampliamente utilizados en el ámbito del aprendizaje informático y, además, en su versión de uso libre, ofrecen una amplia funcionalidad, incluyendo lo ejecutado en el presente análisis. Se obtiene un proceso comparativo real como resultado de las pruebas de las inserciones, consultas y eliminaciones de datos en cada uno de los gestores con comandos definidos en lenguaje SQL (Abad, 2015).

La evaluación de rendimiento permitió, entonces, mostrar de manera clara los datos estudiados, ya que, a su vez, facilitó agruparlos de manera congruente para la elección de un gestor de bases de datos acorde con la consulta que se vaya a realizar.

Se puede concluir que el gestor más funcional para quienes trabajan con una base de datos con

demasiadas tablas y grandes cantidades de datos es Oracle, ya que a la hora de consultar registros que llevan a cabo muchos procesos tiene mejor rendimiento. Para el proceso de agregar datos y realizar pocas consultas, el más eficiente es utilizar Mysql. En últimas, Mysql es más rápido a la hora de insertar o borrar datos, mientras que Oracle es más rápido a la hora de consultar registros.

No obstante, cabe anotar que los análisis y resultados aquí descritos son usados para fines educativos. Esto, porque las características de los gestores descritos generalmente son las que se usan en un aula de clases o para aprendices, los cuales no cuentan con su versión completa.

REFERENCIAS

- Abad, V. (2015). *Estudio comparativo de BDD relacionales y No SQL en un entorno industrial* [Tesis de doctorado]. Escuela Politécnica Nacional, Quito, Ecuador. Recuperado de <https://riunet.upv.es/handle/10251/55530>
- Almonacid Inzunza, D. J. (2016). *Comparación entre gestores de bases de datos relacionales* [Tesis de doctorado]. Universidad Católica de la Santísima Concepción, Chile. Recuperado de <http://repositoriodigital.ucsc.cl/handle/25022009/1092>
- Cardona, G. A., Rendón, F. N., & Reyes, R. (2016). Prototipo para identificación de seguridad en envases de bebidas alcohólicas. *Actas de Ingeniería*, 2, 334-343. Recuperado de <http://fundacioniai.org/actas/Actas2/Actas2.38.pdf>
- Cuervo, M. C. (2012). Evaluación del rendimiento de los motores de bases de datos Mysql y Firebird. *Revista Universidad EAFIT*, 43(148), 78-90. Recuperado de <http://publicaciones.eafit.edu.co/index.php/revista-universidad-eafit/article/view/703>
- Díez del Valle Medrano, Á. & Torreño, F. (2016). *Comparativa del rendimiento de consultas entre sistemas relacionales* [Tesis de grado]. Universidad Politécnica, Madrid, España. Recuperado de <http://oa.upm.es/42375/>
- García Frutos, R. (2017). *Optimización de consultas en bases de datos relacionales* [Tesis de grado]. Universidad Carlos III, Madrid, España. Recuperado de <http://hdl.handle.net/10016/24295>
- Gómez Ruiz, S. (2007). Microsoft SQL Server, Mysql y Postgre SQL. *IPMagazine*. Recuperado de https://www.protaia.com/articulos/52_53_54_55_56_57.pdf
- HR Schema (2003). *Oracle Database Sample Schemas*. Recuperado de https://docs.oracle.com/cd/B13789_01/server.101/b10771/scripts003.htm
- López Herrera, P. (2016). *Comparación del desempeño de los Sistemas Gestores de Bases de Datos Mysql y Postgre SQL* [Tesis de grado]. Universidad Autónoma del Estado de México. Recuperado de <http://ri.uaemex.mx/handle/20.500.11799/62548>
- Mysql. (2017). *Límites en varias bases de datos y tablas*. Recuperado de <https://dev.mysql.com/doc/refman/5.5/en/database-count-limit.html>
- Oracle. (2017). *Database Reference: Datatype Limits*. Recuperado de http://docs.oracle.com/cd/B19306_01/server.102/b14237/limits001.htm#i287903
- Oracle Universit. (s. f). *Oracle Database 12c: SQL Fundamentals*. Recuperado de http://education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=5001&get_params=p_exam_id:1Z0061#exam_preparation&p_org_id=1000258&lang=ESA
- Pilco Guachi, L. (2012). *Estudio Comparativo entre Sistemas de Gestion de Bases de Datos Libres FIREBIRD y MYSQL en el Desarrollo de Aplicaciones Web* [Tesis de grado]. Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador. Recuperado de <http://dspace.esepoch.edu.ec/handle/123456789/1534>
- Torres, L., López, F, & Flores, P. (2016). Evaluador de sentencias de bases de datos en la formación de ingenieros. *Anfei 5*. Recuperado de <http://www.anfei.org.mx/revista/index.php/revista/article/view/288/929>
- Zhingri, V., & Augusto, C. (2016). *Análisis de rendimiento entre la base de datos relacional: Mysql y una base de datos no relacional: MongoDB* [Tesis de grado]. Universidad del Azuay, Cuenca, Ecuador. Recuperado de <http://dspace.uazuay.edu.ec/handle/datos/5319>