



6. BANCOS DE PRUEBAS.

6.1. METODOLOGÍA A EMPLEAR.

En este apartado, se van a realizar una serie de pruebas a las bases de datos en estudio, para determinar varios factores que puedan dar una idea del rendimiento del sistema gestor de bases de datos en cuestión ante situaciones determinadas. Para ello se utilizarán diversos bancos de pruebas existentes:

- Crash-me
- MySQL Benchmark Suite
- SysBench

Los dos primeros bancos de pruebas vienen incluidos en cualquier distribución de MySQL, mientras que el tercero es un banco de pruebas independiente desarrollado por terceras partes. Estos bancos de pruebas permitirán probar diversas características de cada sistema gestor de base de datos. Un poco más adelante se entrará más en profundidad sobre los aspectos fundamentales de cada uno.

6.1.1. Ambiente de las pruebas.

6.1.1.1. Igualdad de recursos.

Con el fin de que las pruebas se realicen en un entorno de total igualdad para ambos sistemas gestores de bases de datos, se utilizarán los mismos recursos físicos y lógicos ejecutándose la prueba en la misma máquina, con el mismo sistema operativo, y configurando cada uno con la misma cantidad de utilización recursos lógicos.

6.1.1.2. Recursos físicos.

Partiendo de las características mínimas de instalación que requiere para un buen funcionamiento cada uno de los dos sistemas gestores de bases de datos a estudio, se realizarán las pruebas en un equipo de mejores recursos físicos, para asegurar la estabilidad del sistema para todos los test que se van a ejecutar. De este modo, los recursos físicos empleados en el desarrollo de las pruebas han sido los siguientes:

- Procesador Intel Pentium IV HT a 3.0 GHz.
- 1024 MB de memoria RAM.
- Disco duro con aprox. 20 GB de espacio libre para la realización de las pruebas.

Esta configuración de hardware asegura, según los fabricantes de ambos sistemas gestores de bases de datos, el correcto funcionamiento de cada uno.

6.1.1.3. Recursos lógicos.

El software que se ha empleado para la realización de las pruebas se muestra en el siguiente listado:

- Sistema Operativo Linux. Distribución FEDORA CORE 4. Versión kernel 2.6.11.



- MySQL v4.1.11.
- PostgreSQL v8.1.3.
- Sysbench v0.3.1.
- Lenguaje de programación Perl para ejecutar los bancos de pruebas escritos en dicho lenguaje.

Por otro lado, la configuración adoptada para cada uno de los sistemas operativos (en MySQL se puede ver con el comando `SHOW VARIABLES` y en PostgreSQL con el comando `SHOW ALL`) se puede consultar en los archivos llamados “variables_sistema.txt” que se adjuntan en el CDRom que acompaña a este documento. En estos archivos se ha almacenado el resultado de redireccionar la salida estándar al ejecutar los comandos `SHOW` anteriores.

6.1.1.4. Igualdad en el diseño.

Para la realización de las pruebas se emplearon las mismas opciones de configuración de los bancos de pruebas para cada sistema gestor de base de datos a estudio, de forma que dicho banco de pruebas cree bases de datos con las mismas características, inserte el mismo volumen de información, etc.

6.1.2. Concurrencia.

De los tres bancos de prueba que se van a emplear, *Sysbench* soporta el funcionamiento multiusuario, de forma que es capaz de simular el rendimiento del sistema para diversos valores de carga de usuario. Ello se empleará para evaluar el comportamiento concurrente de cada uno de los sistemas cuando reciben peticiones de varios usuarios simultáneamente. Para ello se hará una batería de pruebas con diversos números de usuarios simultáneos, con un límite máximo marcado por los recursos físicos y lógicos del sistema.

Por defecto, tanto MySQL como PostgreSQL vienen configurados con un determinado número máximo de usuarios en el sistema. Para MySQL este número tiene un valor de 100, y para PostgreSQL dicho valor por defecto es 32. Por tanto, para realizar las pruebas con más usuarios hay que modificar la variable del sistema correspondiente.

6.2. CRASH-ME.

6.2.1. Descripción.

Esta aplicación se trata de un *script* escrito en lenguaje Perl que viene incluido en la distribución de MySQL. Por tanto, para ejecutarlo se deberá tener instalado Perl en la máquina en la que se quiera ejecutar. Este programa intenta encontrar todos los límites y capacidades para un determinado servidor ejecutando una serie de consultas. A través de este test se puede obtener información sobre qué tipo de columna soporta el servidor, el número máximo de índices que pueden existir, las funciones que soporta, el tamaño máximo de una consulta, el tamaño máximo para cada tipo de dato, etc.

Para ejecutar este *script*, estando activo el servidor MySQL, habrá que ejecutar la sentencia siguiente desde la línea de comandos de Linux:



perl crash-me [opciones]

Las opciones soportadas son las mostradas en la tabla 6.1.

Opción	Efecto
--help	Muestra ayuda sobre el comando.
--batch-mode	No requiere confirmación para la ejecución y aborta la operación en caso de error.
--config-file='nombre_fichero'	Lee los resultados desde un fichero específico.
--comment='comentario'	Añade algún comentario al archivo de texto generado.
--check-server	Chequea el servidor antes de conectar. Si no está listo, intenta una nueva conexión.
--database='nombre_bd'	Crea las tablas para la prueba en la base de datos indicada. Por defecto se usa la base de datos test.
--dir='directorio_salida'	Directorio donde se almacenarán los ficheros con los resultados de los test.
--debug	Inicia el <i>script</i> en modo depuración.
--fix-limit-file	Da un nuevo formato al fichero de salida obtenido. No se ejecuta ninguna prueba.
--force	No se pide confirmación para ejecutar el test.
--log-all-queries	Imprime todas las consultas que se ejecutan en el test.
--log-all-queries-to-file='nombre_fichero'	Escribe todas las consultas que se hacen en el test en un fichero.
--host='nombre_host'	Nombre de host. Por defecto se asume <i>localhost</i> .
--user='nombre_usuario'	Nombre del usuario.
--password='password'	Clave del usuario.
--restart	Guarda los estados intermedios durante la ejecución de las pruebas. Esto hace posible continuar el test en el caso de que hubiese existido algún fallo en los <i>drivers</i> DBI o DBD que causaran una salida anómala del programa.
--server='nombre_servidor'	Ejecuta el test sobre el servidor dado. Soporta los servidores Access, Adabas, Empress, Oracle, Informix, DB2, Mimer, mSQL, MS-SQL, MySQL, PostgreSQL, Solid y Sybase.
--suffix='sufijo'	Añade el sufijo especificado al fichero de salida.
--db-start-cmd='comando'	Automáticamente reinicia el servidor con este comando si dicho servidor muere durante la ejecución del test.
--sleep='tiempo'	Espera una cantidad de tiempo, especificada en segundos, antes de reiniciar el servidor.
--verbose	Se almacenan en el fichero de resultados las consultas ejecutadas para la determinar el valor de los parámetros.
--noverbose	

Tabla 6.1. Opciones de *crash-me*.

Al ejecutar el comando anterior con las opciones adecuadas, se pedirá una confirmación (si no se indica lo contrario en las opciones). Esta medida de precaución es porque este test lleva “al límite” al servidor, y existe una pequeña probabilidad de que se cree un estado de inestabilidad en el mismo. Al aceptar dicha confirmación, se ejecutará el test. Todas las consultas que ejecuta el test cumplen el estándar SQL, por lo que cualquier servidor SQL razonable no debería tener problemas para ejecutar estas pruebas. Como resultado se obtendrán un fichero con toda la información de las pruebas realizadas. Este fichero se llama *nombre_servidor[-sufijo].cfg*, que almacena una traza de todas las consultas ejecutadas, con sus correspondientes resultados. Además, se muestra por pantalla un resumen de la ejecución del test con toda la información de las características soportadas y no soportadas por el servidor.

6.2.2. Resultados obtenidos.

En la ejecución de las pruebas, la salida por pantalla se ha redireccionado hacia un fichero de texto, para poder presentar los resultados en este documento adecuadamente. Las tablas 6.2 a 6.13 muestran los resultados obtenidos en la ejecución de dicho banco de pruebas para cada uno de los sistemas gestores de bases de datos a estudio. En dicha tabla, aparecerán sombreadas aquellas filas que contengan diferencias significativas entre cada uno de los sistemas.

Sintaxis SQL	MySQL	PostgreSQL
Tablas sin clave primaria	Si	Si
SELECT sin FROM	Si	Si
SELECT constantes	Si	Si
SELECT nombre_tabla.*	Si	Si
Permite ' y " como delimitadores de cadena	Si	No
Doble " como ' en cadenas	Si	Si
Cadenas de múltiples líneas	Si	Si
" como identificador (ANSI SQL)	error	No
` como identificador	Si	No
[] como identificador	No	No
Doble "" en identificadores como "	No	Si
Alias de columnas	Si	Si
Alias de tablas	Si	Si
Funciones	Si	Si
Funciones de grupo	Si	Si
Funciones de grupo con DISTINCT	Si	Si
Funciones de grupo con varios DISTINCT	Si	Si
Group by	Si	Si
Group by position	Si	Si
Group by alias	Si	Si
Group on unused column	Si	Si
Order by	Si	Si
Order by position	Si	Si
Order by function	Si	Si
Order by on unused column	Si	Si
Compute	No	No
INSERT con lista de valores	Si	No
INSERT con sintaxis SET	Si	No
INSERT con DEFAULT	Si	Si
INSERT con lista de valores vacíos	No	No
INSERT DEFAULT VALUES	No	Si
Permite terminar con ';'	Si	Si
LIMIT número de filas	con LIMIT	con LIMIT
SELECT con LIMIT #,#	Si	No
SELECT con LIMIT # OFFSET #	Si	Si
Alter table add column	Si	Si
Alter table add many columns	Si	Con add
Alter table change column	Si	No
Alter table modify column	Si	No
Alter table alter column default	Si	Si
Alter table drop column	Si	Si
Alter table rename table	Si	Si
Truncate	Si	Si
Alter table add constraint	Si	Si
Alter table drop constraint	No	Si
Alter table add unique	Si	Si



Sintaxis SQL	MySQL	PostgreSQL
Alter table drop unique	con drop key	con constraint
Alter table add primary key	con constraint	con constraint
Alter table add foreign key	Si	Si
Alter table drop foreign key	con drop foreign key	con drop constraint
Alter table drop primary key	drop primary key	drop constraint
Comparaciones Case insensitive	Si	No
Ignorar el espacio final en comparaciones	Si	Si
Group en columnas con valores NULL	Si	Si
Having	Si	Si
Having con funciones de grupo	Si	Si
Order by alias	Si	Si
Having on alias	Si	No
Números binarios	No	No
Números hexadecimales (0x41)	Si	No
Cadenas binarias (b'0110')	No	Si
Cadenas hexadecimales (x'l'ace')	Si	Si
Valor de la operación lógica (1=1)	1	1
Valor de TRUE	1	1
Valor de FALSE	0	0
Mezcla de integer y float en expresión	Si	Si
No necesidad de cast de integer a float	Si	Si
1+NULL = NULL	Si	Si
concat('a',NULL) = NULL	Si	Si
NOROUND	No	No
Soporta formato YYYY-MM-DD (ISO)	Si	Si
Soporta formato DATE 'YYYY-MM-DD' (ISO)	Si	Si
Soporta formato DD.MM.YYYY (EUR)	error	No
Soporta formato DATE 'DD.MM.YYYY' (EUR)	error	No
Soporta formato YYYYMMDD	Si	Si
Soporta formato DATE 'YYYYMMDD'	Si	Si
Soporta formato MM/DD/YYYY	error	Si
Soporta formato DATE 'MM/DD/YYYY'	error	Si
Soporta fechas 0000-00-00	Si	No
Soporta fechas 0001-01-01	Si	Si
Soporta fechas 9999-12-31	Si	Si
Soporta fechas infinitas	error	No
Soporta fechas YY-MM-DD	Si	No
WEEK	USA	No
Soporta formato de hora HH:MM:SS (ISO)	Si	Si
Soporta formato de hora HH.MM.SS (EUR)	error	No
Soporta formato de hora HHHHmmSS	Si	No
Soporta formato de hora HHmmSS	Si	No
Soporta formato de hora HH:MM:SS (AM PM)	error	Si
LIKE en números	Si	Si
columna LIKE columna	Si	Si
Actualización de column= -column	Si	Si
Funciones de cadena en columnas de fecha	Si	Si
char se rellena con espacios	No	No
Update con muchas tablas	Si	No
DELETE FROM table1,table2...	Si	No
Update con subconsultas	Si	Si
Calcula 1--1	Si	No
JOINS simples ANSI SQL	Si	Si
primary key en create table	Si	Si
unique en create table	Si	Si
unique null en create	Si	Si



Sintaxis SQL	MySQL	PostgreSQL
Valor por defecto para columnas	Si	Si
Valor de función por defecto para columnas	No	Si
Tablas temporales	Si	Si
Crear tablas a partir de SELECT	Si	Con AS
Índices en create table	Si	No
create index	Si	Si
drop index	con 'ON'	Si
null en index	Si	Si
null in unique index	Si	Si
null combination en unique index	Si	Si
Índices en parte de una columna(extensión)	Si	No
Espacio de nombres diferentes para los índices	Si	No
Nombres de tablas case insensitive	No	Si
Nombres de columnas case insensitive	Si	Si
drop table if exists	Si	No
create table if not exists	Si	No
inner join	Si	Si
left outer join	Si	Si
natural left outer join	Si	Si
left outer join using	Si	Si
left outer join odbc style	Si	No
right outer join	Si	Si
full outer join	No	Si
cross join (igual que from a,b)	Si	Si
natural join	Si	Si
Union	Si	Si
union all	Si	Si
Intersect	No	Si
intersect all	No	Si
Except	No	Si
except all	No	Si
Minus	No	No
natural join (listas incompatibles)	Si	Si
union (listas incompatibles)	Si	Si
union all (listas incompatibles)	Si	Si
intersect (listas incompatibles)	No	No
intersect all (listas incompatibles)	No	No
except (listas incompatibles)	No	No
except all (listas incompatibles)	No	No
minus (listas incompatibles)	No	No
Subconsultas	Si	Si
Subconsultas recursivas	+64	+64
insert INTO ... SELECT ...	Si	Si
Transacciones	Si	Si
rollback_metadata	No	Si
Actualizaciones atómicas	No	No
Vistas	No	Si
Claves externas	Sólo sintaxis	Si
Create SCHEMA	No	Si
CHARACTER_LENGTH(varchar_field) =	actual length	actual length
Restricciones de columna	Sólo sintaxis	si
Restricciones de tabla	Sólo sintaxis	si
Restricciones con nombre	Sólo sintaxis	si
Restricciones NULL (estilo SyBase)	Si	Si
Triggers (disparadores) (ANSI SQL)	No	No
Procedimientos PSM (ANSI SQL)	No	No



Sintaxis SQL	MySQL	PostgreSQL
Módulos PSM (ANSI SQL)	No	No
Funciones PSM (ANSI SQL)	No	No
Dominios (ANSI SQL)	No	Si
lock table	Si	Si
Muchas tablas en drop table	Si	Si
drop table con restricción CASCADE	Si	Si
-- como comentario (ANSI)	Si	Si
// como comentario	No	No
# como comentario	Si	No
/* */ como comentario	Si	Si
Insertar cadena vacía	Si	Si
Having con alias	Si	No

Tabla 6.2. Diferencias en la sintaxis SQL

Límites	MySQL	PostgreSQL
Conexiones simultaneas (valor por defecto)	100	32
Tamaño de consulta	1048574	16777216
Máximo tamaño de text o blob	1048543	+8000000
Tamaño de cadena constante en WHERE	1048539	16777181
Tamaño de cadena constante en SELECT	1048565	16777207
Tamaño de cadena devuelto de una función	1048576	+8000000
Expresiones simples	1658	+10000
Expresiones grandes	10	+100
Expresiones apiladas	1658	+2000
OR y AND en WHERE	42830	11907
tablas en JOIN	61	+64
Longitud de nombre de tabla	64	+512
Longitud de nombre de columna	64	+512
Longitud de nombre del alias de SELECT	+512	+512
Longitud de nombre del alias de tabla	+512	+512
Longitud del nombre de índice	64	+512
Tamaño máximo de char()	1048543	+8000000
Tamaño máximo de varchar()	1048543	+8000000
Tamaño máximo de text o blob	1048543 (cache)	+8000000 (cache)
Columnas en tabla	2599	1600
Índices unique	+64	+64
Maximo numero de columnas en indices multicolumna	16	32
Máxima longitud de columna en índice multicolumna	255	235328
Longitud de parte varchar en índice multicolumna	255	235328
Índices	+64	+64
Longitud del índice	1000	+8192
Máxima longitud de filas (sin blobs)	65534	103275
Longitud de filas con NULLs (sin blobs)	65502	103275
Número de columnas en order by	+64	+64
Número de columnas en group by	+64	+64
Aritmetica decimal segura	No	Si
Donde estan los valores NULL en lista de registros ordenados	first	last
Donde estan los valores NULL en lista de registros ordenados (orden inverso)	last	first
Group by siempre ordeados	Si	No
crash-me safe	Si	Si

Tabla 6.3. Valores límite.



Palabras clave	MySQL	PostgreSQL
ABSOLUTE	No	No
ACCESS	No	No
ACTION	No	No
ADD	Si	No
ADMIN	No	No
AFTER	No	No
AGGREGATE	No	No
ALIAS	No	No
ALL	Si	Si
ALLOCATE	No	No
ALTER	Si	No
ANALYZE	Si	Si
AND	Si	Si
ANY	No	Si
ARE	No	No
ARRAY	No	Si
AS	Si	Si
ASC	Si	Si
ASSERTION	No	No
ASYNC	No	No
AT	No	No
AUDIT	No	No
AUTHORIZATION	No	Si
AUTO_INCREMENT	No	No
AVG	No	No
BACKUP	No	No
BDB	No	No
BEFORE	Si	No
BEGIN	No	No
BERKELEYDB	No	No
BETWEEN	Si	Si
BIGINT	Si	No
BINARY	Si	Si
BIT	No	No
BIT_LENGTH	No	No
BLOB	Si	No
BOOLEAN	No	No
BOTH	Si	Si
BREADTH	No	No
BREAK	No	No
BROWSE	No	No
BTREE	No	No
BULK	No	No
BY	Si	No
CALL	No	No
CASCADE	Si	No
CASCADDED	No	No
CASE	Si	Si
CAST	No	Si
CATALOG	No	No
CHANGE	Si	No
CHAR	Si	No
CHARACTER	Si	No
CHARACTER_LENGTH	No	No
CHAR_LENGTH	No	No
CHECK	Si	Si



Palabras clave	MySQL	PostgreSQL
CHECKPOINT	No	No
CLASS	No	No
CLOB	No	No
CLOSE	No	No
CLUSTER	No	No
CLUSTERED	No	No
COALESCE	No	No
COLLATE	Si	Si
COLLATION	No	No
COLUMN	Si	Si
COLUMNS	Si	No
COMMENT	No	No
COMMIT	No	No
COMPLETION	No	No
COMPRESS	No	No
COMPUTE	No	No
CONDITION	No	No
CONNECT	No	No
CONNECTION	No	No
CONSTRAINT	Si	Si
CONSTRAINTS	No	No
CONSTRUCTOR	No	No
CONTAINS	No	No
CONTAINSTABLE	No	No
CONTINUE	No	No
CONVERT	Si	No
CORRESPONDING	No	No
COUNT	No	No
CREATE	Si	Si
CROSS	Si	Si
CUBE	No	No
CURRENT	No	No
CURRENT_DATE	Si	Si
CURRENT_PATH	No	No
CURRENT_ROLE	No	No
CURRENT_TIME	Si	Si
CURRENT_TIMESTAMP	Si	Si
CURRENT_USER	Si	Si
CURSOR	No	No
CYCLE	No	No
DATA	No	No
DATABASE	Si	No
DATABASES	Si	No
DATALINK	No	No
DATE	No	No
DAY	No	No
DAY_HOUR	Si	No
DAY_MINUTE	Si	No
DAY_SECOND	Si	No
DBCC	No	No
DEALLOCATE	No	No
DEC	Si	No
DECIMAL	Si	No
DECLARE	No	No
DEFAULT	Si	Si
DEFERRABLE	No	Si



Palabras clave	MySQL	PostgreSQL
DEFERRED	No	No
DELAYED	Si	No
DELETE	Si	No
DENY	No	No
DEPTH	No	No
DEREF	No	No
DESC	Si	Si
DESCRIBE	Si	No
DESCRIPTOR	No	No
DESTROY	No	No
DESTRUCTOR	No	No
DETERMINISTIC	No	No
DIAGNOSTICS	No	No
DICTIONARY	No	No
DISCONNECT	No	No
DISK	No	No
DISTINCT	Si	Si
DISTINCTROW	Si	No
DISTRIBUTED	No	No
DO	No	Si
DOMAIN	No	No
DOUBLE	Si	No
DROP	Si	No
DUMMY	No	No
DUMP	No	No
DYNAMIC	No	No
EACH	No	No
ELSE	Si	Si
ELSEIF	No	No
ENCLOSED	Si	No
END	No	Si
END-EXEC	Si	Si
EQUALS	No	No
ERRLVL	No	No
ERRORS	No	No
ESCAPE	No	No
ESCAPED	Si	No
EVERY	No	No
EXCEPT	No	Si
EXCEPTION	No	No
EXCLUSIVE	No	No
EXEC	No	No
EXECUTE	No	No
EXISTS	Si	No
EXIT	No	No
EXPAND	No	No
EXPANDING	No	No
EXPLAIN	Si	No
EXTERNAL	No	No
EXTRACT	No	No
FALSE	Si	Si
FETCH	No	No
FIELDS	Si	No
FILE	No	No
FILLFACTOR	No	No
FIRST	No	No



Palabras clave	MySQL	PostgreSQL
FLOAT	Si	No
FOR	Si	Si
FOREIGN	Si	Si
FOUND	No	No
FREE	No	No
FREETEXT	No	No
FREETEXTTABLE	No	No
FROM	Si	Si
FULL	No	Si
FULLTEXT	Si	No
FUNCTION	No	No
GENERAL	No	No
GEOMETRY	No	No
GET	No	No
GLOBAL	No	No
GO	No	No
GOTO	No	No
GRANT	Si	Si
GROUP	Si	Si
GROUPING	No	No
HANDLER	No	No
HASH	No	No
HAST	No	No
HAVING	Si	Si
HIGH_PRIORITY	Si	No
HOLDLOCK	No	No
HOST	No	No
HOUR	No	No
HOUR_MINUTE	Si	No
HOUR_SECOND	Si	No
IDENTIFIED	No	No
IDENTITY	No	No
IDENTITYCOL	No	No
IDENTITY_INSERT	No	No
IF	Si	No
IGNORE	Si	No
IMMEDIATE	No	No
IN	Si	Si
INCREMENT	No	No
INDEX	Si	No
INDICATOR	No	No
INFILE	Si	No
INITIAL	No	No
INITIALIZE	No	No
INITIALLY	No	Si
INNER	Si	Si
INNODB	No	No
INOUT	No	No
INPUT	No	No
INSENSITIVE	No	No
INSERT	Si	No
INT	Si	No
INTEGER	Si	No
INTERSECT	No	Si
INTERVAL	Si	No
INTO	Si	Si



Palabras clave	MySQL	PostgreSQL
IS	Si	Si
ISOLATION	No	No
ITERATE	No	No
JOIN	Si	Si
KEY	Si	No
KEYS	Si	No
KILL	Si	No
LANGUAGE	No	No
LARGE	No	No
LAST	No	No
LATERAL	No	No
LEADING	Si	Si
LEAVE	No	No
LEFT	Si	Si
LESS	No	No
LEVEL	No	No
LIKE	Si	Si
LIMIT	Si	Si
LINENO	No	No
LINES	Si	No
LOAD	Si	No
LOCAL	No	No
LOCALTIME	Si	Si
LOCALTIMESTAMP	Si	Si
LOCATOR	No	No
LOCK	Si	No
LONG	Si	No
LOB	Si	No
LONGTEXT	Si	No
LOOP	No	No
LOWER	No	No
LOW_PRIORITY	Si	No
MASTER_SERVER_ID	No	No
MATCH	Si	No
MAX	No	No
MAXEXTENTS	No	No
MEDIUMBLOB	Si	No
MEDIUMINT	Si	No
MEDIUMTEXT	Si	No
MEETS	No	No
MIDDLEINT	Si	No
MIN	No	No
MINUS	No	No
MINUTE	No	No
MINUTE_SECOND	Si	No
MLSLABEL	No	No
MODE	No	No
MODIFIES	No	No
MODIFY	No	No
MODULE	No	No
MONTH	No	No
MRG_MYISAM	No	No
NAMES	No	No
NATIONAL	No	No
NATURAL	Si	Si
NCHAR	No	No



Palabras clave	MySQL	PostgreSQL
NCLOB	No	No
NEW	No	Si
NEXT	No	No
NOAUDIT	No	No
NOCHECK	No	No
NOCOMPRESS	No	No
NONCLUSTERED	No	No
NONE	No	No
NORMALIZE	No	No
NOT	Si	Si
NOWAIT	No	No
NULL	Si	Si
NULLIF	No	No
NUMBER	No	No
NUMERIC	Si	No
OBJECT	No	No
OCTET_LENGTH	No	No
OF	No	No
OFF	No	Si
OFFLINE	No	No
OFFSETS	No	No
OID	No	Si
OLD	No	Si
ON	Si	Si
ONLINE	No	No
ONLY	No	Si
OPEN	No	No
OPENDATASOURCE	No	No
OPENQUERY	No	No
OPENROWSET	No	No
OPENXML	No	No
OPERATION	No	No
OPERATORS	No	No
OPTIMIZE	Si	No
OPTION	Si	No
OPTIONALLY	Si	No
OR	Si	Si
ORDER	Si	Si
ORDINALITY	No	No
OTHERS	No	No
OUT	No	No
OUTER	Si	Si
OUTFILE	Si	No
OUTPUT	No	No
OVER	No	No
OVERLAPS	No	Si
PAD	No	No
PARAMETER	No	No
PARAMETERS	No	No
PARTIAL	No	No
PATH	No	No
PCTFREE	No	No
PENDANT	No	No
PERCENT	No	No
PERIOD	No	No
PLAN	No	No



Palabras clave	MySQL	PostgreSQL
POSITION	No	No
POSTFIX	No	No
PRECEDES	No	No
PRECISION	Si	No
PREFIX	No	No
PREORDER	No	No
PREPARE	No	No
PRESERVE	No	No
PRIMARY	Si	Si
PRINT	No	No
PRIOR	No	No
PRIVATE	No	No
PRIVILEGES	Si	No
PROC	No	No
PROCEDURE	Si	No
PROTECTED	No	No
PUBLIC	No	No
PURGE	Si	No
RAISERROR	No	No
RAW	No	No
READ	Si	No
READS	No	No
READTEXT	No	No
REAL	Si	No
RECONFIGURE	No	No
RECURSIVE	No	No
REDO	No	No
REF	No	No
REFERENCES	Si	Si
REFERENCING	No	No
REGEXP	Si	No
RELATIVE	No	No
RENAME	Si	No
REPEAT	No	No
REPLACE	Si	No
REPLICATION	No	No
REQUIRE	Si	No
RESIGNAL	No	No
RESOURCE	No	No
RESTORE	No	No
RESTRICT	Si	No
RESULT	No	No
RETURN	No	No
RETURNS	No	No
REVOKE	Si	No
RIGHT	Si	Si
RLIKE	Si	No
ROLE	No	No
ROLLBACK	No	No
ROLLUP	No	No
ROUTINE	No	No
ROW	No	No
ROWCOUNT	No	No
ROWGUIDCOL	No	No
ROWID	No	No
ROWNUM	No	No



Palabras clave	MySQL	PostgreSQL
ROWS	No	No
RTREE	No	No
RULE	No	No
SAVE	No	No
SAVEPOINT	No	No
SCHEMA	No	No
SCROLL	No	No
SEARCH	No	No
SECOND	No	No
SECTION	No	No
SELECT	Si	Si
SENSITIVE	No	No
SEQUENCE	No	No
SESSION	No	No
SESSION_USER	No	Si
SET	Si	No
SETS	No	No
SETUSER	No	No
SHARE	No	No
SHOW	Si	No
SHUTDOWN	No	No
SIGNAL	No	No
SIMILAR	No	Si
SIZE	No	No
SMALLINT	Si	No
SOME	No	Si
SONAME	Si	No
SPACE	No	No
SPATIAL	Si	No
SPECIFIC	No	No
SPECIFICTYPE	No	No
SQL	No	No
SQLCODE	No	No
SQLERROR	No	No
SQLEXCEPTION	No	No
SQLSTATE	No	No
SQLWARNING	No	No
SQL_BIG_RESULT	Si	No
SQL_CALC_FOUND_ROWS	Si	No
SQL_SMALL_RESULT	Si	No
SSL	Si	No
START	No	No
STARTING	Si	No
STATE	No	No
STATIC	No	No
STATISTICS	No	No
STRAIGHT_JOIN	Si	No
STRIPED	No	No
STRUCTURE	No	No
SUBSTRING	No	No
SUCCEEDS	No	No
SUCCESSFUL	No	No
SUM	No	No
SYNONYM	No	No
SYSDATE	No	No
SYSTEM_USER	No	No



Palabras clave	MySQL	PostgreSQL
TABLE	Si	Si
TABLES	Si	No
TEMPORARY	No	No
TERMINATE	No	No
TERMINATED	Si	No
TEST	No	No
TEXTSIZE	No	No
THAN	No	No
THEN	Si	Si
THERE	No	No
TIME	No	No
TIMESTAMP	No	No
TIMEZONE_HOUR	No	No
TIMEZONE_MINUTE	No	No
TINYBLOB	Si	No
TINYINT	Si	No
TINYTEXT	Si	No
TO	Si	Si
TOP	No	No
TRAILING	Si	Si
TRAN	No	No
TRANSACTION	No	No
TRANSLATE	No	No
TRANSLATION	No	No
TREAT	No	No
TRIGGER	No	No
TRIM	No	No
TRUE	Si	Si
TRUNCATE	No	No
TSEQUAL	No	No
TYPE	No	No
TYPES	No	No
UID	No	No
UNDER	No	No
UNDO	No	No
UNION	Si	Si
UNIQUE	Si	Si
UNKNOWN	No	No
UNLOCK	Si	No
UNSIGNED	Si	No
UNTIL	No	No
UPDATE	Si	No
UPDATETEXT	No	No
UPPER	No	No
USAGE	Si	No
USE	Si	No
USER	No	Si
USER_RESOURCES	No	No
USING	Si	Si
VALIDATE	No	No
VALUE	No	No
VALUES	Si	No
VARBINARY	Si	No
VARCHAR	Si	No
VARCHAR2	No	No
VARIABLE	No	No



Palabras clave	MySQL	PostgreSQL
VARYING	Si	No
VIEW	No	No
VIRTUAL	No	No
VISIBLE	No	No
WAIT	No	No
WAITFOR	No	No
WARNINGS	No	No
WHEN	Si	Si
WHENEVER	No	No
WHERE	Si	Si
WHILE	No	No
WITH	Si	No
WITHOUT	No	No
WORK	No	No
WRITE	Si	No
WRITETEXT	No	No
XOR	Si	No
YEAR	No	No
YEAR_MONTH	Si	No
ZEROFILL	Si	No
ZONE	No	No

Tabla 6.4. Palabras clave soportadas.

Tipos SQL soportados	MySQL	PostgreSQL
character(1 arg)	Si	Si
char(1 arg)	Si	Si
char varying(1 arg)	Si	Si
character varying(1 arg)	Si	Si
boolean	Si	Si
varchar(1 arg)	Si	Si
integer	Si	Si
int	Si	Si
smallint	Si	Si
numeric(2 arg)	Si	Si
decimal(2 arg)	Si	Si
dec(2 arg)	Si	Si
bit	Si	Si
bit(1 arg)	Si	Si
bit varying(1 arg)	No	Si
float	Si	Si
float(1 arg)	Si	Si
real	Si	Si
double precision	Si	Si
date	Si	Si
time	Si	Si
timestamp	Si	Si
interval year	No	Si
interval year to month	No	Si
interval month	No	Si
interval day	No	Si
interval day to hour	No	Si
interval day to minute	No	Si
interval day to second	No	Si
interval hour	No	Si
interval hour to minute	No	Si



Tipos SQL soportados	MySQL	PostgreSQL
interval hour to second	No	Si
interval minute	No	Si
interval minute to second	No	Si
interval second	No	Si
national character varying(1 arg)	Si	Si
national character(1 arg)	Si	Si
nchar(1 arg)	Si	Si
national char varying(1 arg)	Si	Si
nchar varying(1 arg)	Si	Si
national character varying(1 arg)	Si	Si
timestamp with time zone	No	Si

Tabla 6.5. Tipos de datos SQL soportados.

Tipos ODBC soportados	MySQL	PostgreSQL
binary(1 arg)	Si	No
varbinary(1 arg)	Si	No
tinyint	Si	No
bigint	Si	Si
datetime	Si	No

Tabla 6.6. Tipos ODBC soportados.

Tipos extra soportados	MySQL	PostgreSQL
blob	Si	No
byte	No	No
long varbinary	Si	No
image	No	No
text	Si	Si
text(1 arg)	Si	No
mediumtext	Si	No
long varchar(1 arg)	No	No
varchar2(1 arg)	No	No
mediumint	Si	No
middleint	Si	No
int unsigned	Si	No
int1	Si	No
int2	Si	Si
int3	Si	No
int4	Si	Si
int8	Si	Si
uint	No	No
money	No	Si
smallmoney	No	No
float4	Si	Si
float8	Si	Si
smallfloat	No	No
float(2 arg)	Si	No
double	Si	No
enum(1 arg)	Si	No
set(1 arg)	Si	No
int(1 arg) zerofill	Si	No
serial	Si	Si
char(1 arg) binary	Si	No
int not null auto_increment	Si	No
abstime	No	Si



Tipos extra soportados	MySQL	PostgreSQL
year	Si	No
datetime	Si	No
smalldatetime	No	No
timespan	No	No
reltime	No	Si
intnot null identity	No	No
box	No	Si
bool	Si	Si
circle	No	Si
polygon	Si	Si
point	Si	Si
line	No	Si
lseg	No	Si
path	No	Si
interval	No	Si
inet	No	Si
cidr	No	Si
macaddr	No	Si
varchar2(1 arg)	No	No
nvarchar2(1 arg)	No	No
number(2 arg)	No	No
number(1 arg)	No	No
number	No	No
long	Si	No
raw(1 arg)	No	No
long raw	No	No
rowid	No	No
mlslabel	No	No
clob	No	No
nclob	No	No
bfile	No	No
Recuerda el espaciado final en char()	No	No
Recuerda el espaciado final en varchar()	No	Si
Almacenamiento de valores flotantes	round	round
para row id	auto_increment	oid
row id automatico	_rowid	No

Tabla 6.7. Tipos extra soportados.

Funciones SQL soportadas	MySQL	PostgreSQL
+, -, * y /	Si	Si
ANSI SQL SUBSTRING	Si	Si
BIT_LENGTH	Si	Si
searched CASE	Si	Si
simple CASE	Si	Si
CAST	Si	Si
CHARACTER_LENGTH	Si	Si
CHAR_LENGTH	error	error
CHAR_LENGTH(constant)	Si	Si
COALESCE	Si	Si
CURRENT_DATE	Si	Si
CURRENT_TIME	Si	Si
CURRENT_TIMESTAMP	Si	Si
EXTRACT	Si	Si
LOCALTIME	Si	Si
LOCALTIMESTAMP	Si	Si



Funciones SQL soportadas	MySQL	PostgreSQL
LOWER	Si	Si
NULLIF con cadenas	Si	Si
NULLIF con números	Si	Si
OCTET_LENGTH	Si	Si
POSITION	Si	Si
TRIM	Si	Si
UPPER	Si	Si
concatenación con	error	Si

Tabla 6.8. Funciones SQL soportadas.

Funciones extra soportadas	MySQL	PostgreSQL
& (and bit a bit)	Si	Si
(or bit a bit)	Si	Si
<< y >> (desplazamientos bit a bit)	Si	Si
<> en SELECT	Si	Si
=	Si	Si
~* (comparación case insensitive)	No	Si
AND y OR en SELECT	Si	Si
AND como '&&'	Si	No
ASCII_CHAR	No	No
ASCII_CODE	No	No
ATN2	No	No
BETWEEN en SELECT	Si	Si
BIT_COUNT	Si	No
CEIL	Si	Si
CHARINDEX	No	No
CHR	No	Si
CONCAT(list)	Si	No
CONVERT	No	No
COSH	No	No
ELT	Si	No
ENCRYPT	Si	No
FIELD	Si	No
FORMAT	Si	No
GETDATE	No	No
GREATEST	Si	No
IF	Si	No
IN en numeros en SELECT	Si	Si
IN en cadenas en SELECT	Si	Si
INITCAP	No	Si
INSTR (sintaxis Oracle)	No	No
INSTRB	No	No
INTERVAL	Si	No
LAST_INSERT_ID	Si	No
LEAST	Si	No
LENGTHB	No	No
LIKE ESCAPE en SELECT	Si	Si
LIKE en SELECT	Si	Si
LN	Si	Si
LOCATE como INSTR	Si	No
LOG(m,n)	Si	Si
LOGN	No	No
LPAD	Si	Si
MOD como %	Si	Si
NOT BETWEEN en SELECT	Si	Si



Funciones extra soportadas	MySQL	PostgreSQL
NOT LIKE in SELECT	Si	Si
NOT as '!' in SELECT	Si	No
NOT in SELECT	Si	Si
ODBC CONVERT	No	No
OR como ' '	Si	No
PASSWORD	Si	No
PASTE	No	No
PATINDEX	No	No
POW	Si	Si
RANGE	No	No
REGEXP en SELECT	Si	No
REPLICATE	No	No
REVERSE	Si	No
ROOT	No	No
ROUND(1 arg)	Si	Si
RPAD	Si	Si
SINH	No	No
STR	No	No
STRCMP	Si	No
STUFF	No	No
SUBSTRB	No	No
SUBSTRING como MID	Si	No
SUBSTRING_INDEX	Si	No
SYSDATE	Si	No
TAIL	No	No
TANH	No	No
TRANSLATE	No	Si
TRIM; Extensión con muchos caracteres	error	Si
TRIM; Extensión subcadena	Si	error
TRUNC	No	Si
UID	No	No
UNIX_TIMESTAMP	Si	No
USERENV	No	No
VERSION	Si	Si
conversión num->string automática	Si	No
conversión string->num automática	Si	Si
concatenación con +	error	No
SUBSTR (2 arg)	Si	Si
SUBSTR (3 arg)	Si	Si
LFILL (3 arg)	No	No
RFILL (3 arg)	No	No
RPAD (4 arg)	No	No
TRIM (1 arg)	Si	Si
TRIM (2 arg)	No	Si
LTRIM (2 arg)	No	Si
RTRIM (2 arg)	No	Si
EXPAND	No	No
REPLACE (2 arg)	No	No
MAPCHAR	No	No
ALPHA	No	No
ASCII en cast de cadenas	error	error
EBCDIC en cast de cadenas	No	No
TRUNC (1 arg)	No	Si
FIXED	No	No
FLOAT	No	No
LENGTH	error	error



Funciones extra soportadas	MySQL	PostgreSQL
INDEX	No	No
MICROSECOND	Si	No
TIMESTAMP	error	No
VALUE	No	No
DECODE	No	No
NUM	No	No
CHR (cualquier tipo a string)	No	error
HEX	Si	No
DAYNAME	Si	No
MONTH	Si	No
MONTHNAME	Si	No
DAYOFMONTH	Si	No
DAYOFWEEK	Si	No
DAYOFYEAR	Si	No
QUARTER	Si	No
YEAR	Si	No
DATEADD	No	No
MDY	No	No
DATEDIFF	No	No
DATENAME	No	No
DATEPART	No	No
DATE_FORMAT	Si	No
FROM_DAYS	Si	No
FROM_UNIXTIME	Si	No
MONTHS_BETWEEN	No	No
PERIOD_ADD	Si	No
PERIOD_DIFF	Si	No
WEEKDAY	Si	No
ADDDATE	Si	No
SUBDATE	Si	No
DATEDIFF (2 arg)	Si	No
WEEKOFYEAR	Si	No
CHAR	No	No
MAKEDATE	Si	No
TO_DAYS	Si	No
ADD_MONTHS	No	No
LAST_DAY	Si	No
DATE	Si	Si
Function DAY	Si	No
HOUR	Si	No
ANSI HOUR	Si	No
MINUTE	Si	No
SECOND	Si	No
TIME_TO_SEC	Si	No
SEC_TO_TIME	Si	No
ADDTIME	Si	No
SUBTIME	Si	No
TIMEDIFF	Si	No
MAKETIME	Si	No
TIME	Si	No
NOT ID BETWEEN se interpreta como ID NOT BETWEEN	No	Si

Tabla 6.9. Funciones extra soportadas.

Funciones de grupo SQL soportadas	MySQL	PostgreSQL
AVG	Si	Si



Funciones de grupo SQL soportadas	MySQL	PostgreSQL
COUNT (*)	Si	Si
COUNT nombre_columna	Si	Si
COUNT(DISTINCT expr)	Si	Si
MAX en números	Si	Si
MAX en cadenas	Si	Si
MIN en números	Si	Si
MIN en cadenas	Si	Si
SUM	Si	Si
ANY	No	No
EVERY	No	No
SOME	No	No

Tabla 6.10. Funciones de grupo SQL soportadas.

Funciones ODBC soportadas	MySQL	PostgreSQL
ASCII	Si	Si
CHAR	Si	No
CONCAT(2 arg)	Si	No
DIFFERENCE()	No	No
INSERT	Si	No
LEFT	Si	No
LTRIM	Si	Si
REAL LENGTH	Si	Si
ODBC LENGTH	error	No
LOCATE(2 arg)	Si	No
LOCATE(3 arg)	Si	No
LCASE	Si	No
REPEAT	Si	Si
REPLACE	Si	Si
RIGHT	Si	No
RTRIM	Si	Si
SPACE	Si	No
SOUNDEX	Si	No
ODBC SUBSTRING	Si	Si
UCASE	Si	No
ABS	Si	Si
ACOS	Si	Si
ASIN	Si	Si
ATAN	Si	Si
ATAN2	Si	Si
CEILING	Si	Si
COS	Si	Si
COT	Si	Si
DEGREES	Si	Si
EXP	Si	Si
FLOOR	Si	Si
LOG	Si	No
LOG10	Si	No
MOD	Si	Si
PI	Si	Si
POWER	Si	Si
RAND	Si	No
RADIANS	Si	Si
ROUND(2 arg)	Si	Si
SIGN	Si	Si
SIN	Si	Si



Funciones ODBC soportadas	MySQL	PostgreSQL
SQRT	Si	Si
TAN	Si	Si
TRUNCATE	Si	No
NOW	Si	Si
CURDATE	Si	No
CURTIME	Si	No
TIMESTAMPADD	No	No
TIMESTAMPDIFF	No	No
USER()	Si	No
DATABASE	Si	No
IFNULL	Si	No
LEFT & RIGHT	Si	No

Tabla 6.11. Funciones ODBC soportadas.

Funciones WHERE soportadas	MySQL	PostgreSQL
= ALL	Si	Si
= ANY	Si	Si
= SOME	Si	Si
BETWEEN	Si	Si
EXISTS	Si	Si
IN en números	Si	Si
LIKE ESCAPE	Si	Si
LIKE	Si	Si
MATCH UNIQUE	No	No
MATCH	No	No
MATCHES	No	No
noT BETWEEN	Si	Si
noT EXISTS	Si	Si
noT LIKE	Si	Si
noT UNIQUE	No	No
UNIQUE	No	No

Tabla 6.12. Funciones WHERE soportadas.

Funciones de grupo adicionales soportadas	MySQL	PostgreSQL
BIT_AND	Si	Si
BIT_OR	Si	Si
COUNT(DISTINCT expr,expr,...)	Si	No
STD	Si	No
STDDEV	Si	Si
VARIANCE	Si	Si

Tabla 6.13. Funciones de grupo adicionales soportadas.

6.2.3. Comentarios.

Como puede verse en el listado anterior, el conjunto de resultados de esta prueba es bastante extenso, pues se muestran todas las características soportadas y no soportadas por cada uno de los sistemas gestores de bases de datos. Si se echa un vistazo al contenido de las tablas, sobre todo a las filas sombreadas, se aprecian las diferencias entre cada uno de los sistemas gestores de bases de datos, algunas de ellas bastante significativas. Lo primero que se puede concluir es que existen algunas diferencias no muy notables en la sintaxis SQL, lo cual queda patente si se examina el listado de



palabras clave soportadas. No obstante, en general, ambos sistemas se adaptan bien al estándar SQL.

En este listado quedan patentes algunas de las diferencias que se comentaron anteriormente en el análisis teórico comparativo de ambos sistemas, tales como las relativas a transacciones, vistas, restricciones, etc.

Del análisis de los resultados anteriores se puede concluir que MySQL resulta beneficiado en los siguientes aspectos:

- Mayor número de conexiones simultaneas.
- Tipos de datos numéricos más amplio.
- Número de funciones nativas sensiblemente superior.
- Mayor número de condiciones OR/AND en consultas.
- Índices en un espacio de nombres diferente.
- Creación/eliminación de tablas si no existen/existen.
- Mayor número de columnas en una tabla.

Por el contrario, PostgreSQL sale beneficiado en los siguientes aspectos:

- Mayor tamaño máximo de consulta.
- Tipos de datos más extensos (a excepción de los numéricos).
- Mayor tamaño máximo de los tipos de datos.
- Permite cadenas de mayor tamaño en consultas.
- Permite mayor tamaño en cadenas devueltas por funciones.
- Permite consultas JOIN con mayor número de tablas.
- Valores de función por defecto en columnas.
- Nombre de tablas no sensible a mayúsculas/minúsculas.
- Sentencias Intersect/Except.
- Permite mayor longitud en nombres de tablas, columnas e índices.
- Permite mayor número de columnas en índices multicolumnas.
- Permite índices de mayor longitud.
- Posee aritmética decimal segura (mas precisa).

Por tanto, a partir de los resultados anteriores, y de los obtenidos en el análisis teórico comparativo se puede concluir que PostgreSQL resulta ser un sistema gestor de bases de datos que soporta un mayor número de características y funcionalidades que MySQL.

6.3. MySQL BENCHMARK SUITE.

6.3.1. Descripción.

MySQL viene con su propia suite de herramientas de banco de pruebas, Esta suite incluye la aplicación Crash-me, analizada en el apartado anterior, y una herramienta denominada MySQL Bench. Al igual que la aplicación Crash-me, MySQL Bench consiste en un *script* escrito en lenguaje Perl, por lo que se deberá tener instalado dicho lenguaje de programación en el sistema. MySQL Bench sirve tanto para analizar las diferencias de rendimiento entra diferentes versiones de MySQL como para comparar el rendimiento con respecto a otros sistemas gestores de bases de datos.



Este banco de pruebas resulta muy útil para ver tiempos de respuesta de cualquier sistema gestor de bases de datos ante gran variedad de pruebas y operaciones. Sin embargo, posee una gran carencia, y es que no permite realizar análisis multi-usuario, es decir, que con esta herramienta no se podrán realizar pruebas de concurrencia en las diferentes bases de datos. Para este tipo de pruebas, se empleará otro tipo de banco de pruebas, cuyo análisis y resultados se verán en el siguiente apartado. No obstante, según el equipo de desarrollo de MySQL, se pretende dotar a MySQL Bench de soporte multiusuario en un futuro.

Para ejecutar este banco de pruebas, el comando que se deberá emplear es el siguiente:

```
perl run-all-tets [opciones]
```

Este *script* se deberá ejecutar desde el mismo directorio en que se encuentra (habitualmente /usr/share/sql-bench), o bien especificar la ruta completa en el comando. Al ejecutar este *script* se realizarán un total de 9 pruebas:

- alter-table
- ATIS
- big-tables
- connect
- create
- insert
- select
- transactions
- wisconsin

Al final del test se obtendrá el resultado de la ejecución del test. Este resultado se mostrará por pantalla o se escribirá en un archivo de salida determinado. Además, también se generará un archivo por cada prueba, que mostrará información detallada sobre cada una de ellas (acciones realizadas, tiempo para cada acción, etc.). Los tiempos mostrados se muestran en segundos “*wallclock*”, y después de cada valor se especifica la cantidad de tiempo empleada por el *script* para las tareas que no están directamente relacionadas con la prueba.

Existen algunas opciones para modificar el comportamiento del *script* o para especificar algún aspecto adicional. Estas opciones se muestran en la tabla 6.14.

Opción	Efecto
--server= <i>nombre_servidor</i>	Especifica el servidor de bases de datos contra el que se van a efectuar las pruebas. Los posibles valores son 'MySQL' (por defecto), 'MS-SQL', 'Oracle', 'DB2', 'mSQL', 'Pg', 'Solid', 'Sybase', 'Adabas', 'Access', 'Empress' e 'Informix'.
--log	Almacena el resultado de las pruebas en un fichero de texto.
--dir= <i>directorio</i>	Especifica el directorio en el que se almacenarán los datos. Por defecto es el directorio output.
--use-old-results	Sobrescribe resultados de test anteriores.
--coment	Sirve para añadir algún comentario al archivo de salida.
--fast	Permite ejecutar comandos SQL fuera del estándar ANSI. De esta forma las consultas son más rápidas.
--host= <i>nombre_host</i>	Especifica el nombre de la máquina donde se está ejecutando el servidor. Por defecto se toma <i>localhost</i> .
--small-test	Con esta opción sólo se hace un test muy simple para asegurar únicamente que MySQL funciona correctamente.
--user= <i>nombre_usuario</i>	Nombre de usuario de la base de datos.
--password= <i>password</i>	Clave del usuario de la base de datos.
--comments	Añade algún comentario a la salida del banco de pruebas.



Opción	Efecto
--cmp= <i>server</i> [, <i>server</i> ,...]	Ejecuta las pruebas con los límites de un determinado servidor.
--create-options= <i>argumento</i>	Argumento extra para todas las sentencias de creación de tablas.
--database= <i>nombre_bd</i>	Especifica el nombre de la base de datos en la que se crean las tablas necesarias para las pruebas. Por defecto se emplea la base de datos test.
--debug	Muestra información para depuración.
--fast-insert	Emplea INSERT INTO <i>nombre_tabla</i> VALUES (...) en vez de INSERT INTO <i>nombre_tabla</i> (...) VALUES (...).
--field-count	Especifica el número de campos que deben contener las tablas.
--force	Fuerza la ejecución completa del test, aunque haya error.
--groups	Especifica cuántos grupos debería haber en las pruebas. (Se emplea para depuración)
--loop-count	Especifica cuántas veces se ejecuta cada bucle (por defecto 10000). Se emplea para depuración.
--help	Muestra ayuda sobre este <i>script</i> .
--machine=" <i>máquina</i> ó <i>nombreSO</i> "	Especifica la máquina/SO que se añadirá al fichero de salida de las pruebas.
--odbc	Usa el <i>driver</i> ODBC DBI para conectar a la base de datos.
--socket=" <i>dirección</i> "	Especifica la dirección del <i>socket</i> de Unix si la base de datos soporta este tipo de conexión.
--silent	No presenta información acerca del servidor cuando comienza la prueba.
--skip-delete	No elimina las tablas creadas durante la ejecución de las pruebas.
--skip-test= <i>test1</i> [, <i>test2</i> ,...]	No se ejecuta el test especificado.
--suffix= <i>sufijo</i>	Sufijo que se le añadirá a los nombres de los ficheros generados por las pruebas.
--random	Ejecuta las pruebas en orden aleatorio.
--threads=#	Número de Hilos para pruebas multiusuario. Actualmente carece de utilidad, pero se incluye para futuro soporte de pruebas multiusuario.
--tcp-ip	Indica que se esta empleando una conexión tcp/ip.
--time-limit	Tiempo máximo permitido para cada bucle de pruebas, en segundos. Por defecto 600.
--use-old-results	Toma los resultados de pruebas previas, en vez de ejecutar los <i>tests</i> .
--verbose	Muestra información detallada de lo que está sucediendo en cada momento.
--optimization= <i>'comentarios'</i>	Añade comentarios acerca de la optimización del sistema.
--hw= <i>'comentarios'</i>	Añade comentarios sobre el hardware empleado.
--connect-options= <i>'opciones'</i>	Opciones adicionales de conexión.

Tabla 6.14. Opciones de MySQL Benchmark Suite

Los tests de este banco de pruebas tardan bastante tiempo en ejecutarse (del orden de unos 30 minutos a 6 horas, dependiendo de la máquina). Al finalizar las pruebas, se obtendrá un resultado global, resultados de cada operación y resultados detallados de cada uno de las 9 pruebas realizadas.

6.3.2. Resultados obtenidos.

A continuación se muestran los resultados obtenidos al ejecutar las pruebas en los sistemas gestores de base de datos a estudio. En el caso de MySQL se han hecho dos tipos de pruebas, una con el motor de almacenamiento MyISAM (no soporta transacciones) y otra para el motor de almacenamiento InnoDB (soporta transacciones), para ver cómo afecta el soporte de control transaccional al rendimiento de la base de datos.



Lo primero que se va a mostrar es el tiempo empleado en cada una de las pruebas para cada sistema gestor de bases de datos. Los resultados se muestran en las tablas 6.15, 6.16 y 6.17.

MySQL (MyISAM)						
Prueba	Tiempo empleado (seg)	Tiempo empleado en tareas del sistema que no tienen relación con las pruebas (seg)				Tiempo efectivo
		usr	sys	cur	csys	
Alter-table	48	0.08	0.03	0.00	0.00	47,89
ATIS	21	7.07	2.01	0.00	0.00	11,92
Big-tables	19	5.31	4.17	0.00	0.00	9,52
Connect	109	42.15	18.11	0.00	0.00	48,74
Create	163	9.90	1.21	0.00	0.00	151,89
Insert	2012	462.71	101.19	0.00	0.00	1448,1
Select	362	47.63	8.25	0.00	0.00	306,12
Transactions	No se pudo ejecutar el test					---
Wisconsin	7	2.47	0.78	0.00	0.00	3.75

Tabla 6.15. Tiempo empleado en ejecutarse Mysql Bench en MySQL (MyISAM)

MySQL (InnoDB)						
Prueba	Tiempo empleado (seg)	Tiempo empleado en tareas del sistema que no tienen relación con las pruebas (seg)				Tiempo efectivo
		usr	sys	cur	csys	
Alter-table	273	0.08	0.02	0.00	0.00	272.9
ATIS	23	7.39	2.22	0.00	0.00	13.39
Big-tables	24	5.39	4.48	0.00	0.00	14.13
Connect	118	41.45	19.24	0.00	0.00	57.31
Create	364	10.45	1.82	0.00	0.00	351.73
Insert	43541	500.44	108.94	0.00	0.00	42931.62
Select	408	48.25	9.00	0.00	0.00	350.75
Transactions	27	4.46	1.19	0.00	0.00	21.35
Wisconsin	23	2.90	0.88	0.00	0.00	19,22

Tabla 6.16. Tiempo empleado en ejecutarse Mysql Bench en MySQL (InnoDB)

PostgreSQL						
Prueba	Tiempo empleado (seg)	Tiempo empleado en tareas del sistema que no tienen relación con las pruebas (seg)				Tiempo efectivo
		usr	sys	cur	csys	
Alter-table	3	0.65	0.06	0.00	0.00	2.29
ATIS	No se pudo ejecutar el test					---
Big-tables	75	4.56	0.27	0.00	0.00	70.17
Connect	431	169.19	20.39	0.00	0.00	241.42
Create	649	24.22	2.39	0.00	0.00	622.39
Insert	73809.7	807.97	74.42	0.00	0.00	72927.31



PostgreSQL						
Select	1566.48	62.40	5.18	0.00	0.00	1498.89
Transactions	73	24.16	2.39	0.00	0.00	46.45
Wisconsin	44	10.95	0.93	0.00	0.00	32.12

Tabla 6.17. Tiempo empleado en ejecutarse Mysql Bench en PostgreSQL.

En las tablas anteriores se muestra el tiempo total empleado en la ejecución de las pruebas. Este valor incluye porciones de tiempo utilizadas para tareas del sistema operativo, que nada tienen que ver con las pruebas. Estas porciones de tiempo se especifican en las cuatro columnas correspondientes, y habrá que restarlas al tiempo total para mostrar el tiempo efectivo empleado por el test, que se muestra en la última columna. Si se echa un vistazo a las tablas, puede verse que los test se ejecutan más rápido para MySQL empleando el motor de almacenamiento MyISAM, siendo este tiempo considerablemente inferior al de las otras dos configuraciones empleadas.

Cabe comentar que para el sistema gestor MySQL con motor de almacenamiento MyISAM se obtuvo un error en la prueba “*transacciones*”, al no soportar dicho motor de almacenamiento el uso de transacciones. También se obtuvo error al ejecutar el test ATIS sobre PostgreSQL, debido a la sintaxis de determinadas consultas empleadas por dicho test, no soportada por dicho sistema gestor de bases de datos.

A continuación se van a mostrar una serie de gráficas en las que se muestra el resultado de las pruebas anteriores sobre algunas tareas básicas, obtenidas a partir de los datos que devuelve el resumen del banco de pruebas. El listado completo del contenido de cada uno de los ficheros obtenidos se mostrará en el ANEXO A.

Tareas de conexión a la base de datos:

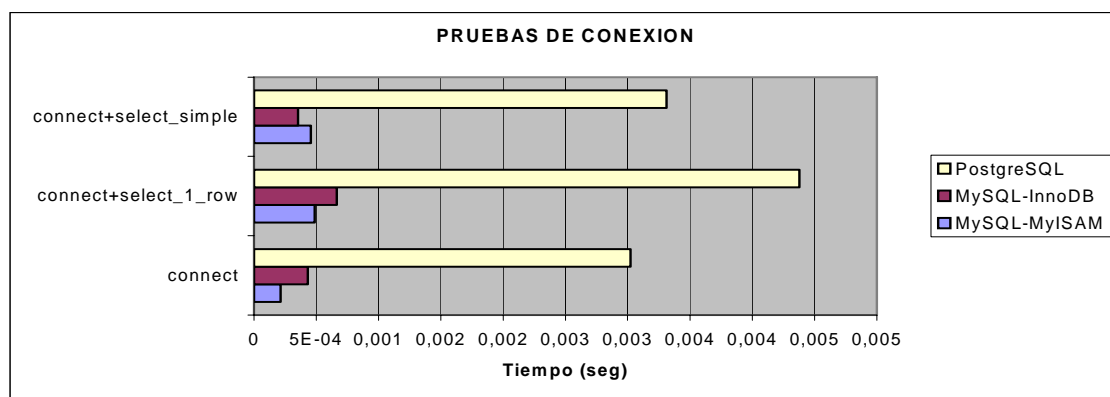


Figura 6.1. Tiempos de conexión a la base de datos.



Tareas de creación de tablas:

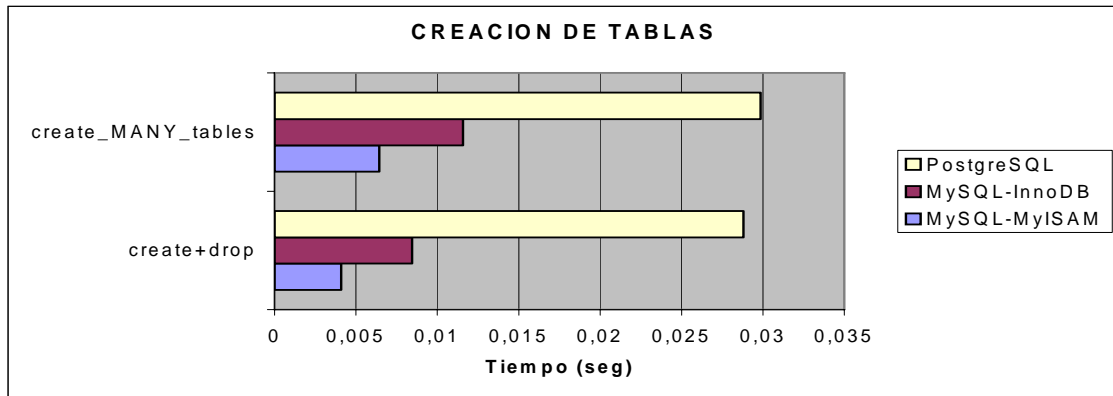


Figura 6.2. Tiempo empleado en tareas de creación de tablas.

Pruebas con la función COUNT():

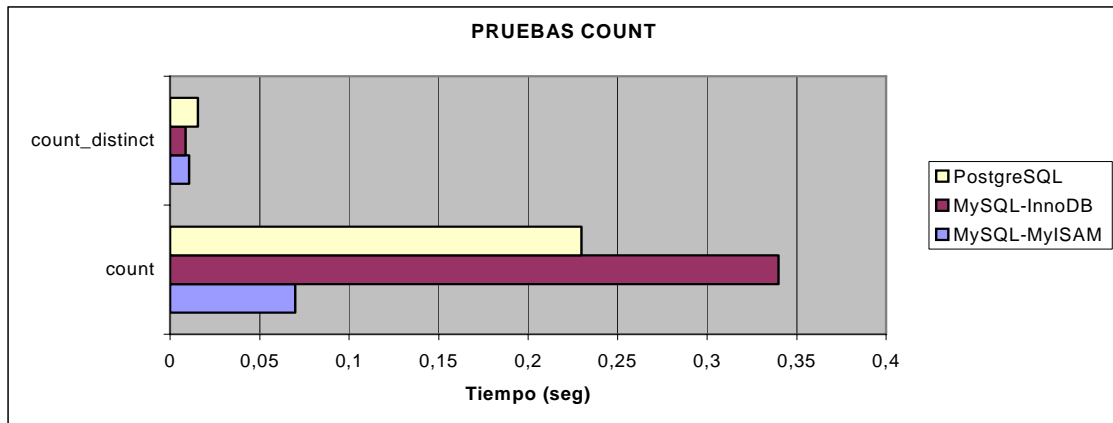


Figura 6.3. Tiempo empleado en tareas que involucran a la función COUNT() (I).

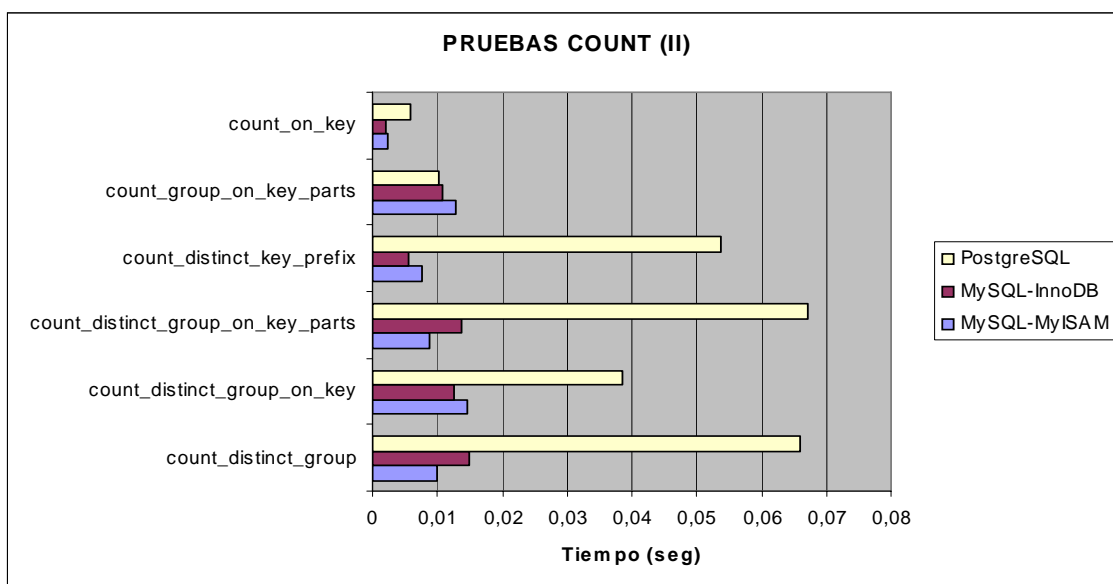


Figura 6.4. Tiempo empleado en tareas que involucran a la función COUNT() (II).



Borrado de elementos:

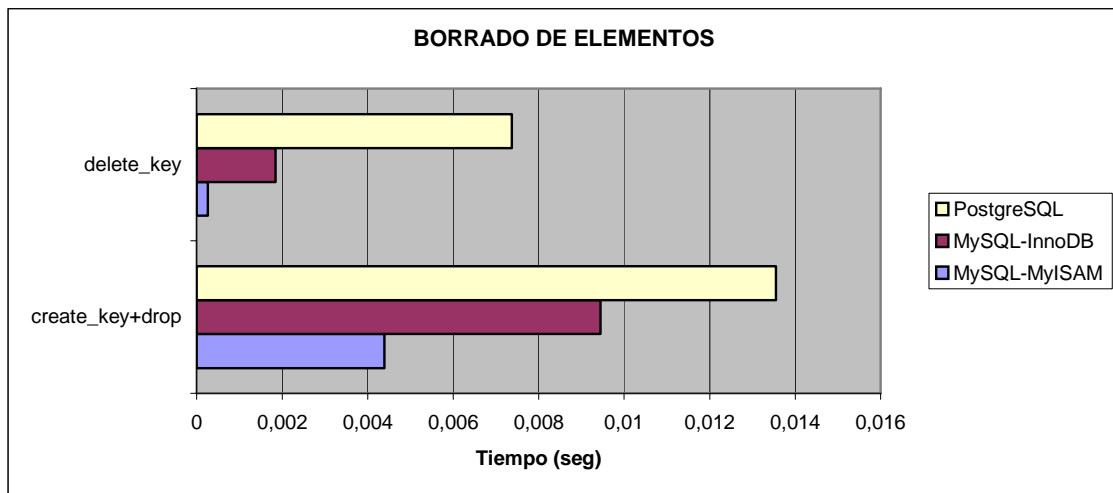


Figura 6.5. Tiempo empleado en tareas de borrado de elementos (I).

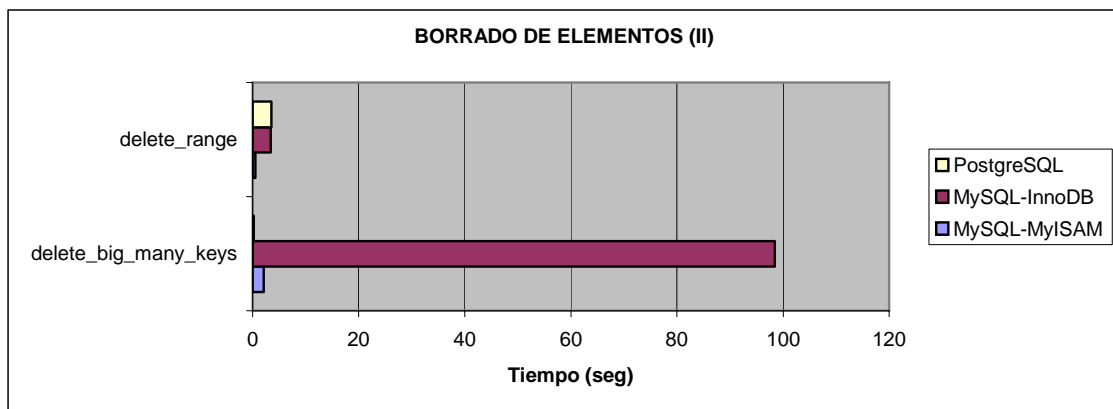


Figura 6.6. Tiempo empleado en tareas de borrado de elementos (II).

Eliminación de índice:

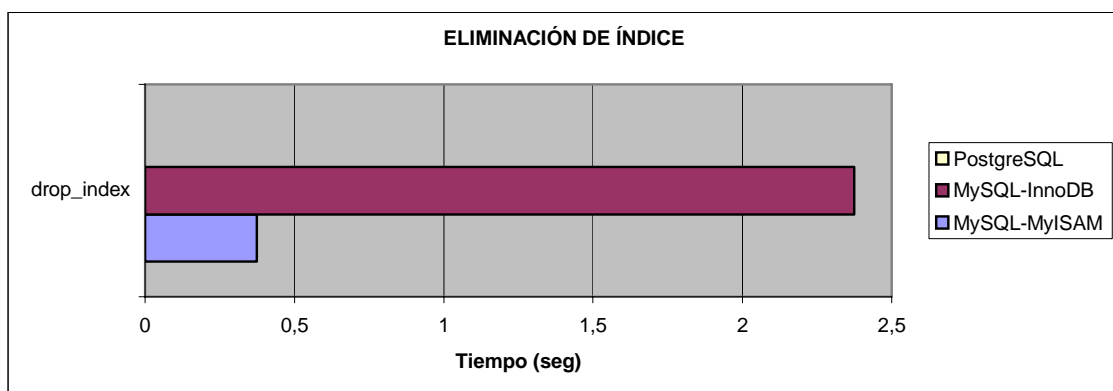


Figura 6.7. Tiempo empleado en la eliminación de un índice.



Inserción de datos:

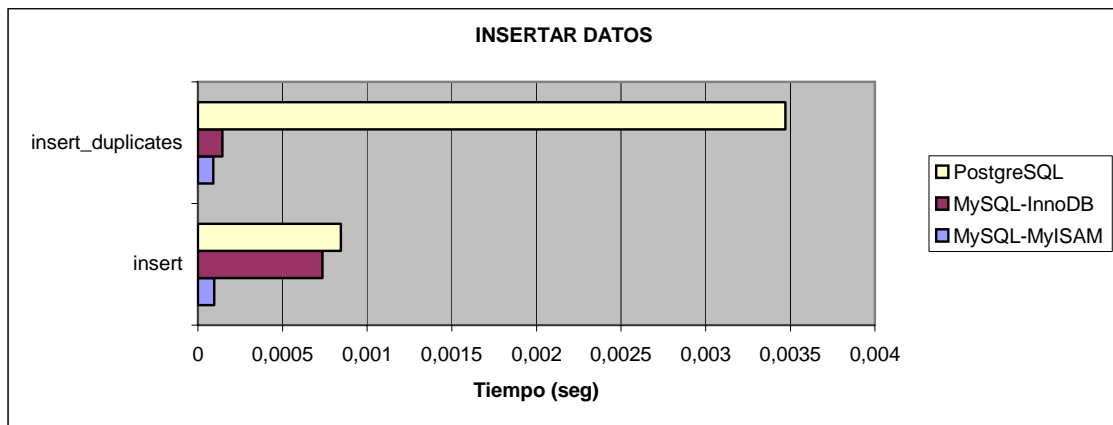


Figura 6.8. Tiempo empleado en tareas de inserción de datos (I).

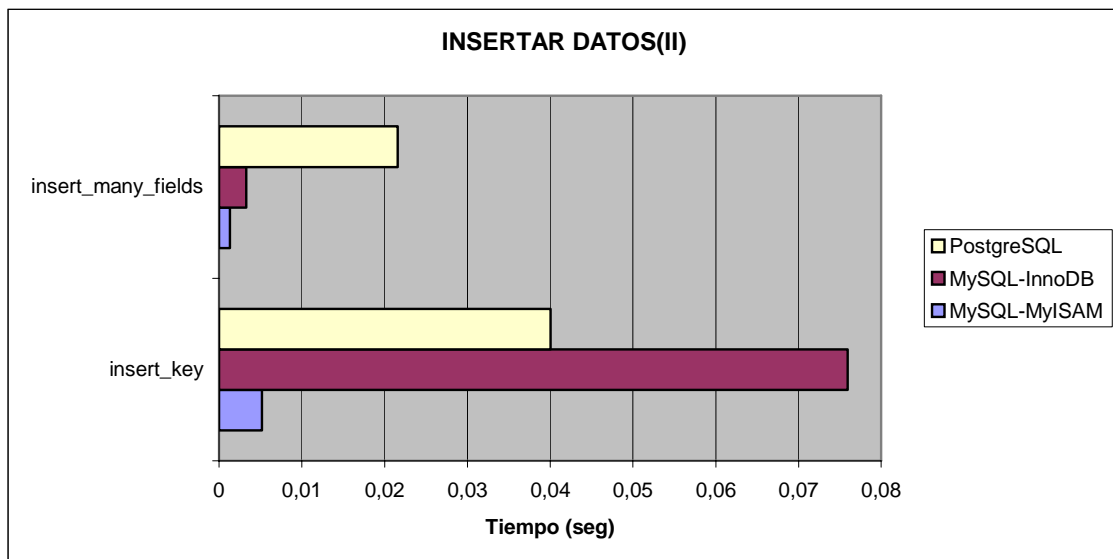


Figura 6.9. Tiempo empleado en tareas de inserción de datos (II).

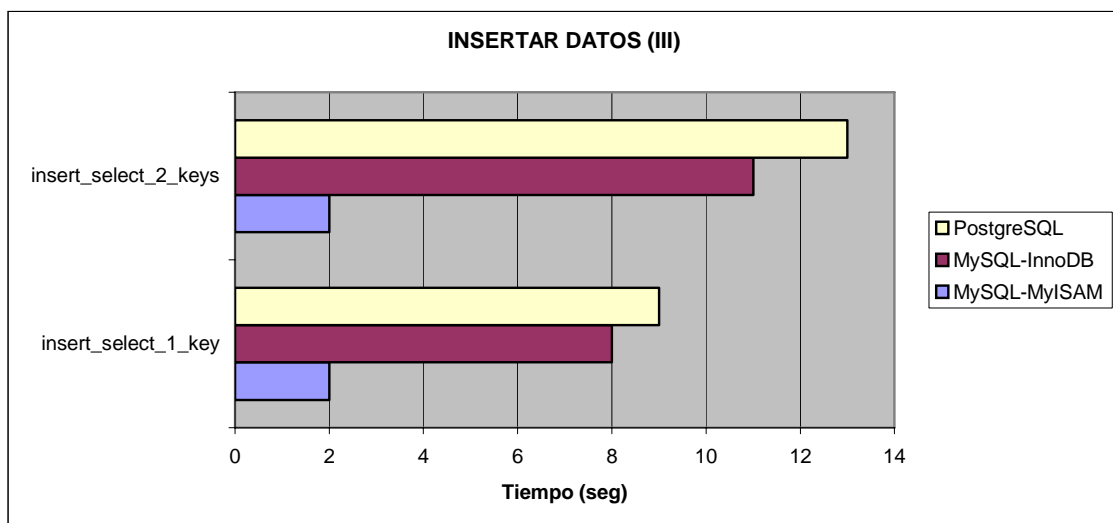


Figura 6.10. Tiempo empleado en tareas de inserción de datos (III).



Consultas SELECT:

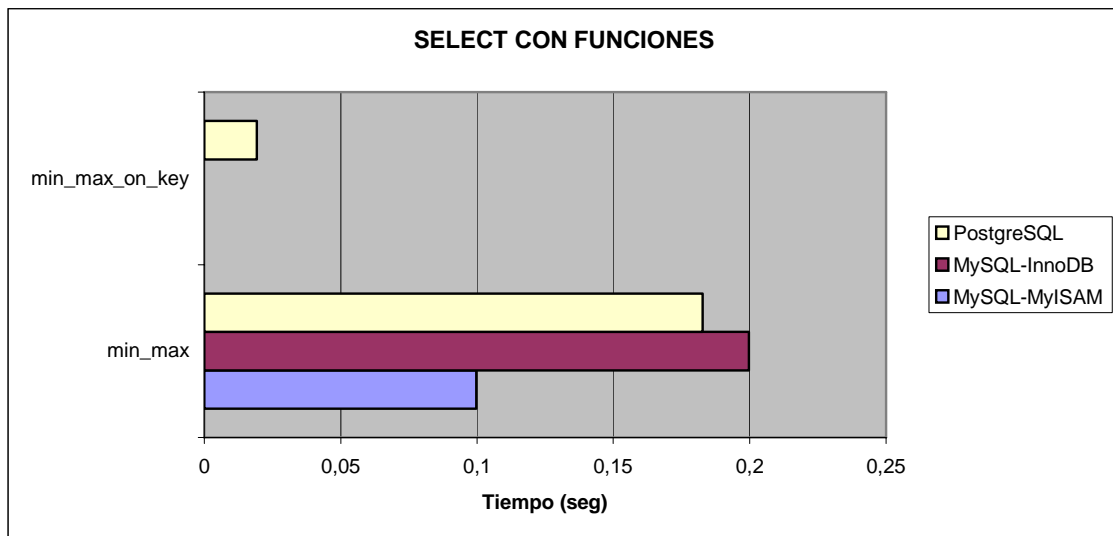


Figura 6.11. Tiempo empleado en tareas que implican consultas de selección con funciones.

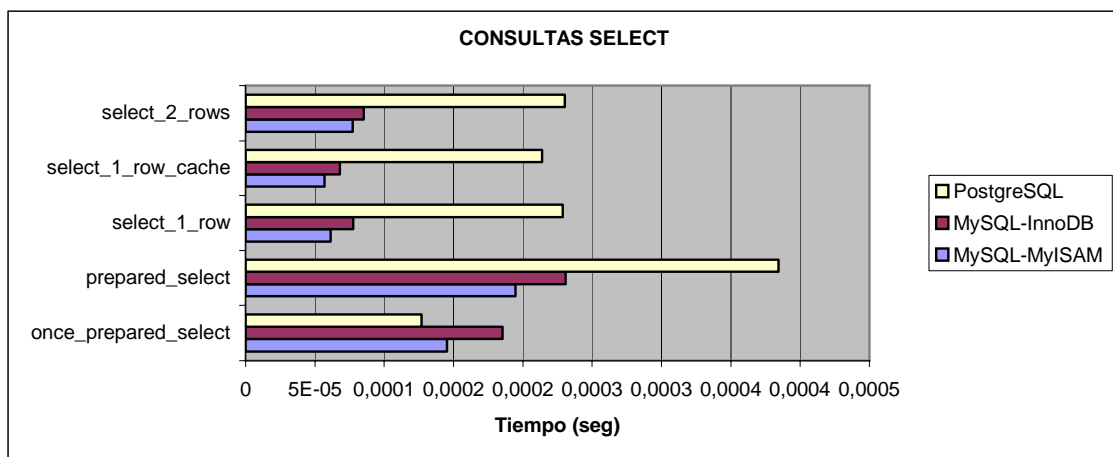


Figura 6.12. Tiempo empleado en tareas que implican consultas de selección (I).

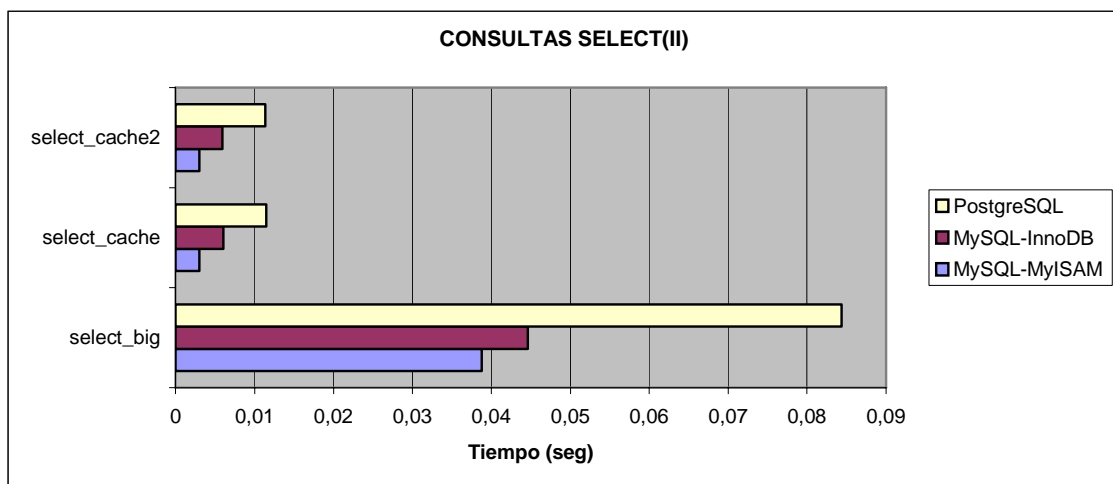


Figura 6.13. Tiempo empleado en tareas que implican consultas de selección (II).

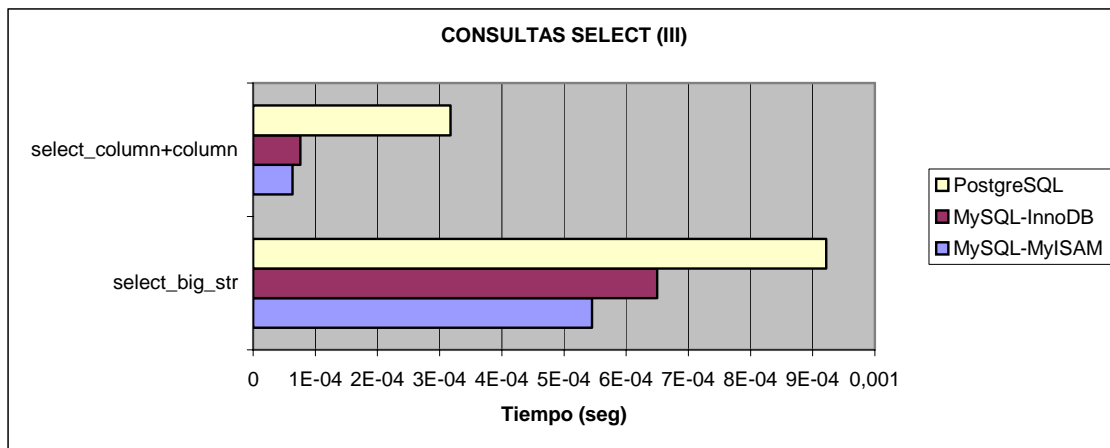


Figura 6.14. Tiempo empleado en tareas que implican consultas de selección (III).

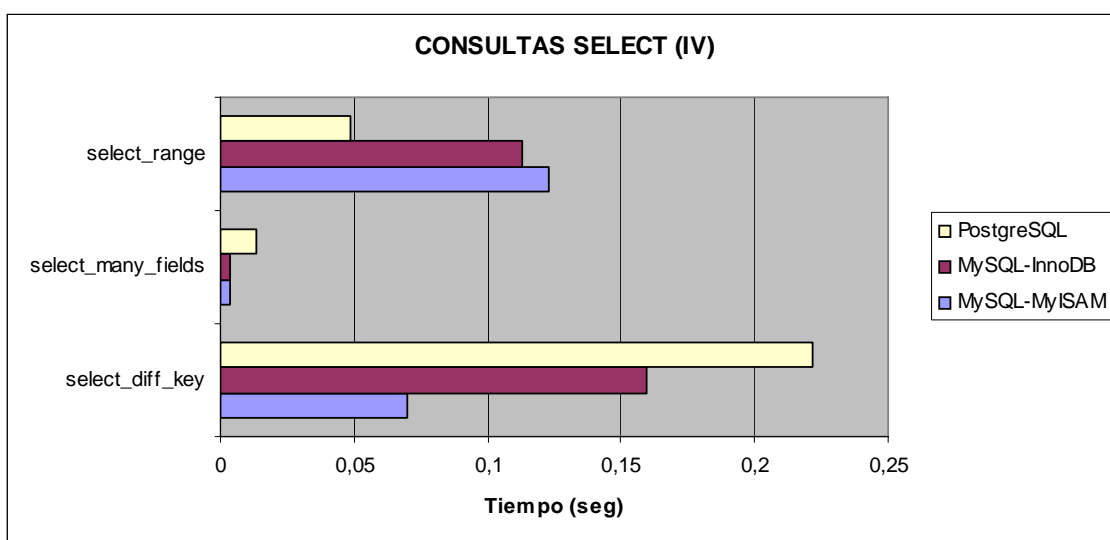


Figura 6.15. Tiempo empleado en tareas que implican consultas de selección (IV).

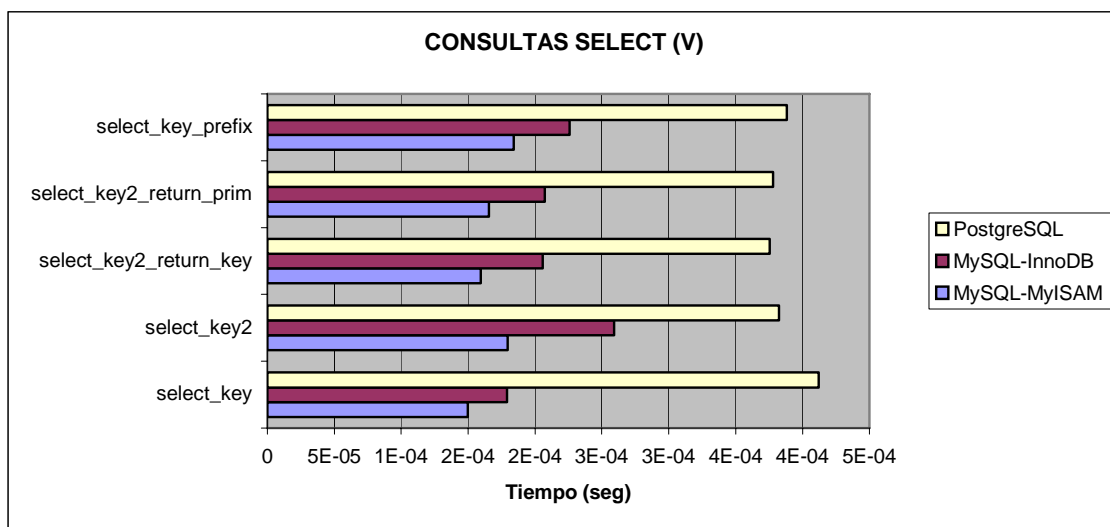


Figura 6.16. Tiempo empleado en tareas que implican consultas de selección (V).

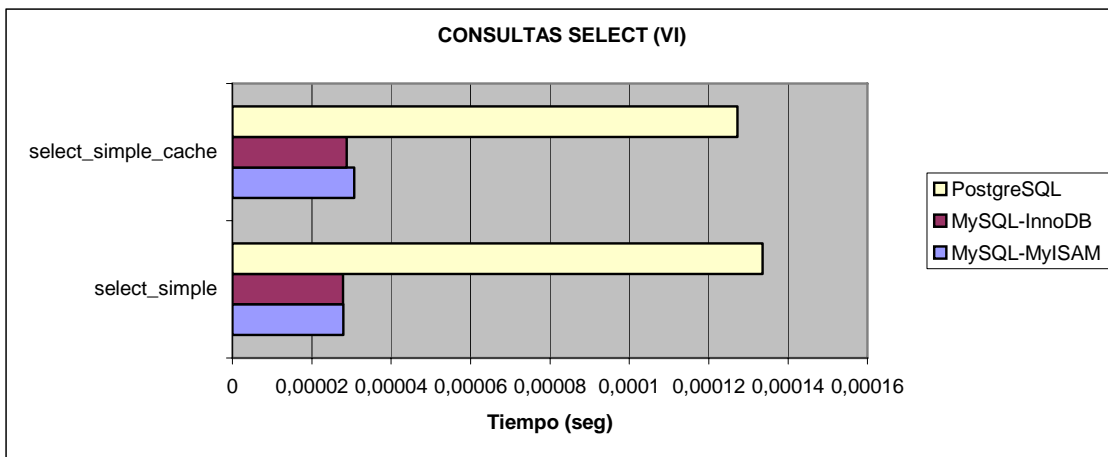


Figura 6.17. Tiempo empleado en tareas que implican consultas de selección (VI).

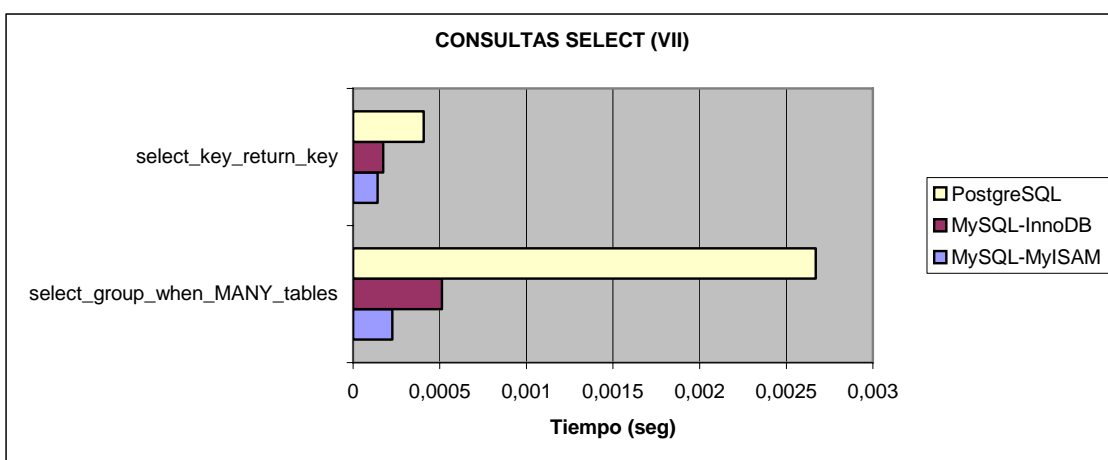


Figura 6.18. Tiempo empleado en tareas que implican consultas de selección (VII).

Consultas ordenadas:

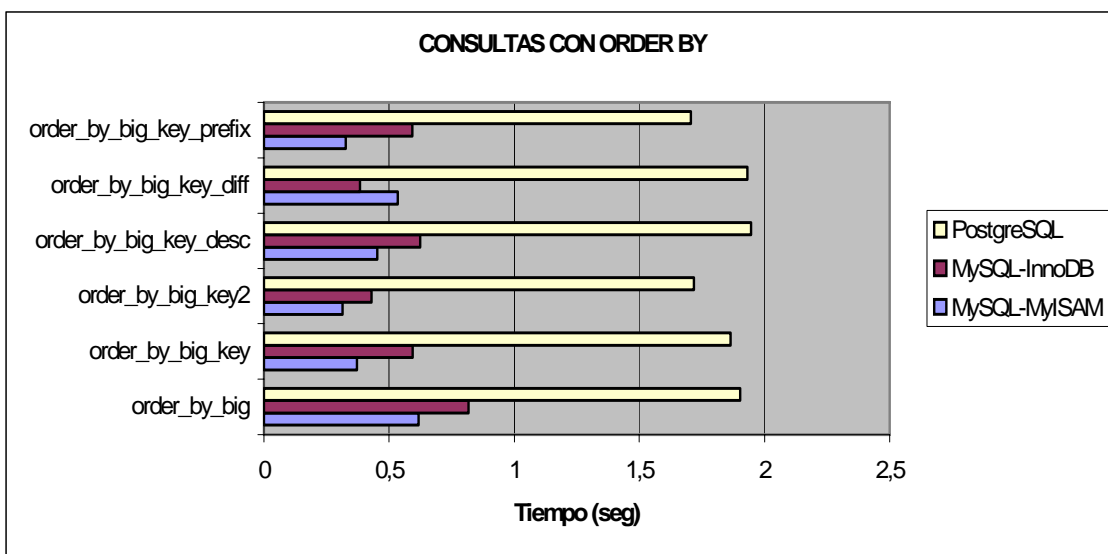


Figura 6.19. Tiempo empleado en consultas con resultados ordenados (I).

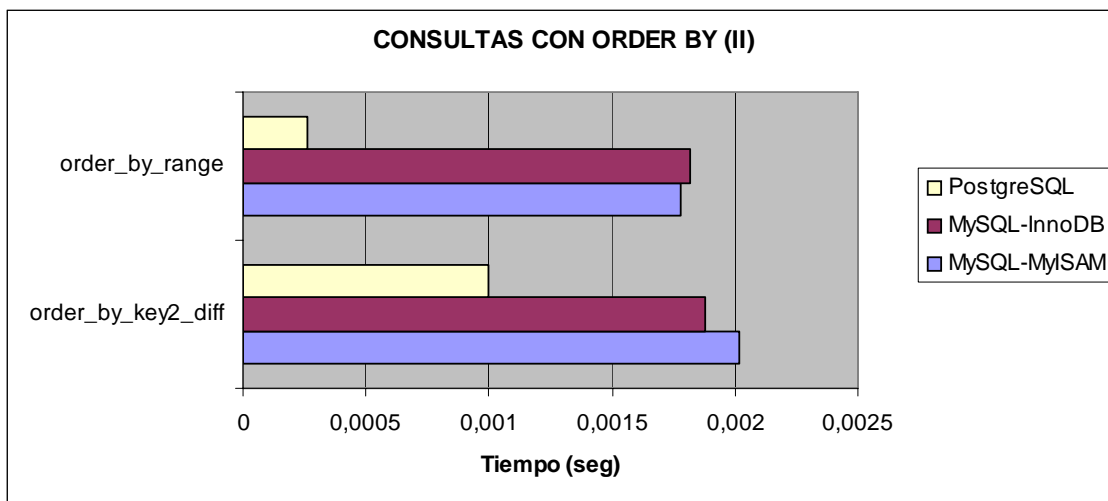


Figura 6.20. Tiempo empleado en consultas con resultados ordenados (II).

Consultas JOIN:

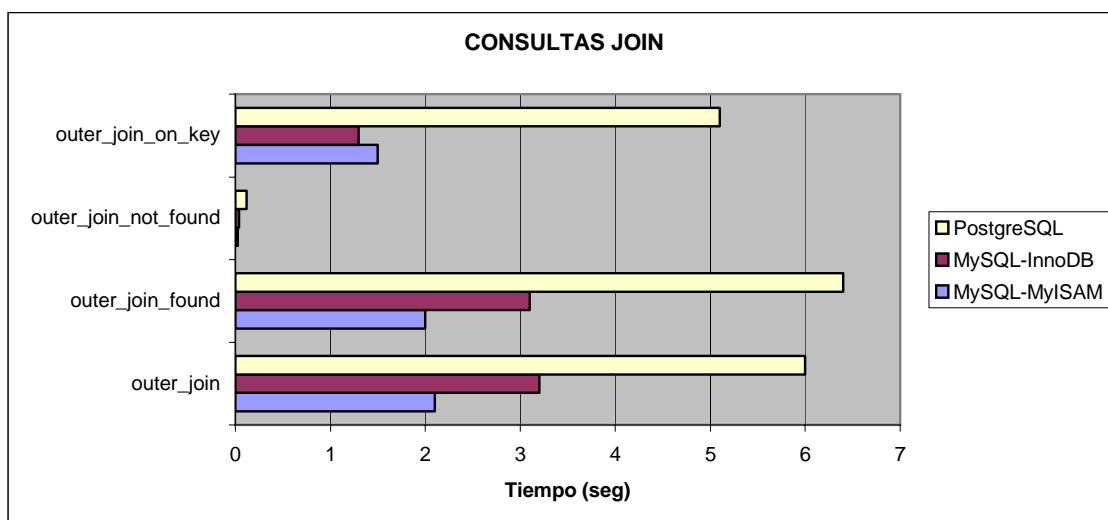


Figura 6.21. Tiempo empleado en consultas JOIN.

Consultas UPDATE:

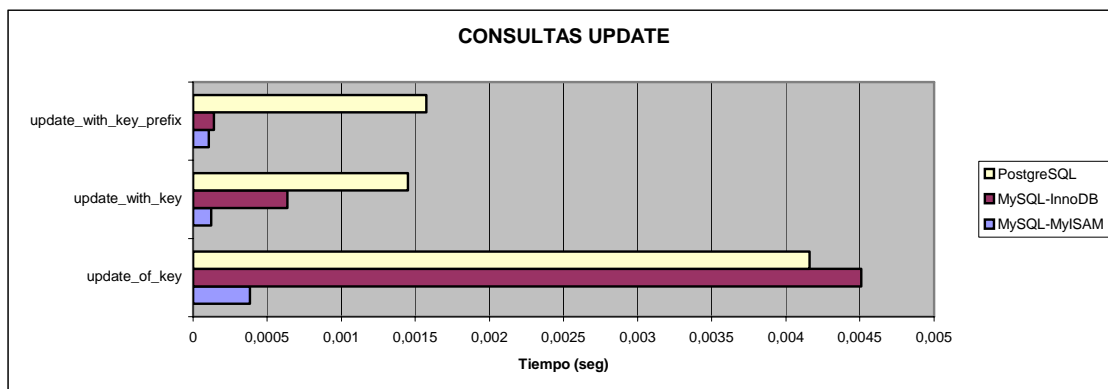


Figura 6.22. Tiempo empleado en consultas UPDATE.



Test Wisconsin:

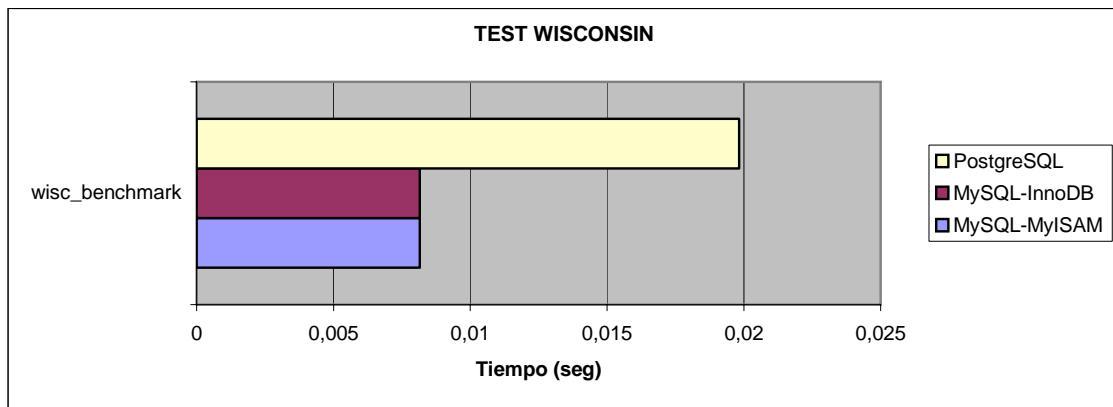


Figura 6.23. Tiempo empleado en realizar el test Wisconsin.

Transacciones:

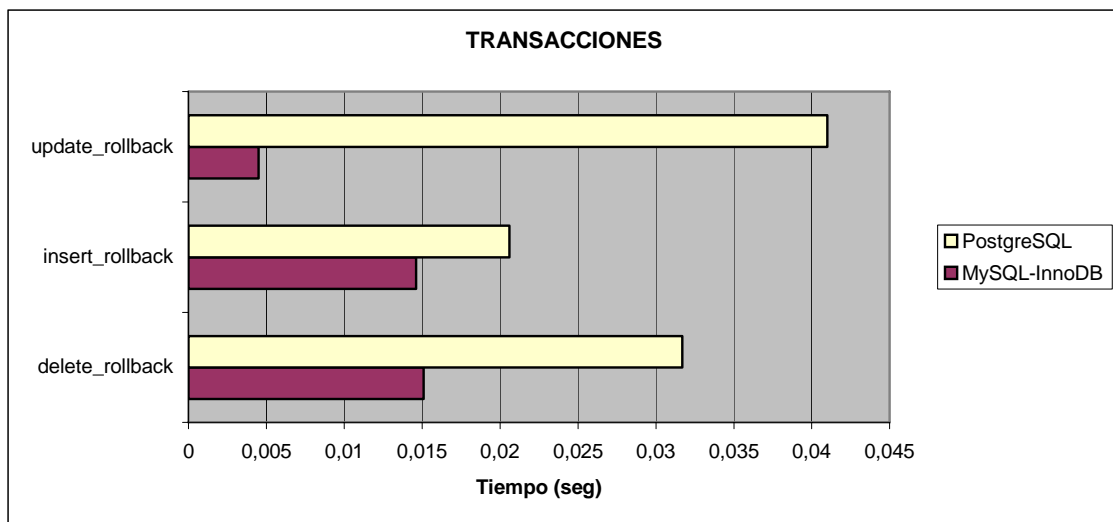


Figura 6.24. Tiempo empleado en tareas que involucran transacciones (I).

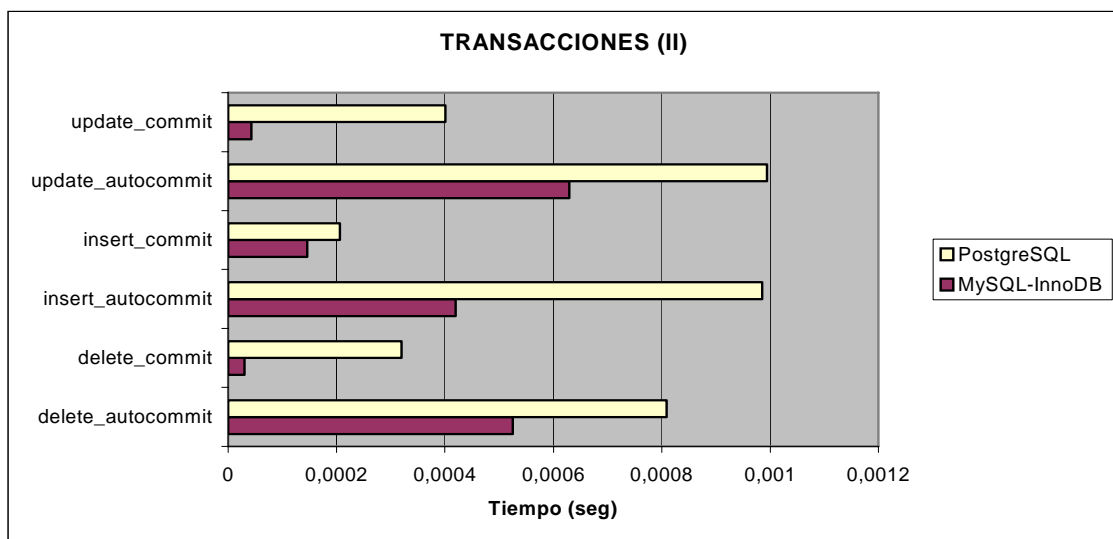


Figura 6.25. Tiempo empleado en tareas que involucran transacciones (II).



6.3.3. Comentarios.

En las gráficas anteriores se puede observar que el sistema gestor de bases de datos que ha obtenido un mayor rendimiento en todas las pruebas ha sido MySQL con el motor de almacenamiento MyISAM, ya que los tiempos de respuesta a la mayoría de las operaciones han sido muy inferiores al de los otros dos casos (MySQL con InnoDB y PostgreSQL). El único caso en que PostgreSQL ha obtenido un rendimiento aceptable comparado con MySQL ha sido en determinadas operaciones que involucran índices, como por ejemplo las consultas ordenadas según un rango. Excluyendo estos casos aislados, la tónica general es que los tiempos de respuesta sean inferiores cuando se emplea MySQL.

En cuanto a los dos casos a estudio que soportan transacciones, se puede ver que el rendimiento de PostgreSQL es bastante más pobre que el de MySQL con motor de almacenamiento InnoDB. Si se comparan estos dos casos, es decir, MySQL con motor de almacenamiento InnoDB y PostgreSQL, se observa que el primero obtiene mayor rendimiento con tiempos de respuesta inferiores en la mayoría de los casos.

Ante todo esto, se pueden obtener varias conclusiones. La primera es que MySQL resulta ser el sistema gestor de bases de datos más rápido de los estudiados, sobre todo con el motor de almacenamiento MyISAM (seleccionado por defecto). Los otros dos casos analizados resultan tener un peor rendimiento, debido al soporte para el control transaccional y otras funcionalidades que aportan, que hace que las consultas sean sensiblemente más lentas. La segunda conclusión es que de entre los dos casos analizados que soportan transacciones (MySQL con InnoDB y PostgreSQL) MySQL muestra unos tiempos de respuesta más bajos. En este sentido, PostgreSQL nota las consecuencias de todas las funcionalidades y características que MySQL no aporta.

Las pruebas anteriores han mostrado los tiempos de respuesta de los sistemas gestores de base de datos a estudio ante diversas operaciones. Estas pruebas se han hecho en un entorno de usuario único. En el apartado siguiente se verán las pruebas de concurrencia, para así estudiar el rendimiento de los sistemas cuando se aumenta la cara de usuarios conectados concurrentemente en el sistema.

6.4. SYSBENCH.

6.4.1. Descripción.

Sysbench es un banco de pruebas multiusuario para evaluar los parámetros más importantes de un sistema en el que se está ejecutando un servidor de bases de datos. Este banco de pruebas no sólo evalúa el rendimiento de la base de datos, sino también del sistema. Actualmente permite verificar los siguientes aspectos:

- Lectura-escritura de ficheros.
- Planificador de procesos del sistema operativo.
- Almacenamiento en memoria y velocidad de transferencia.
- Rendimiento de la implementación de los hilos en sistemas POSIX (*Portable Operating System Interface*).
- Rendimiento del servidor de bases de datos (test *oltp*).



De todas las utilidades que proporciona este software, la que realmente interesa para el cometido de este proyecto es la última, es decir, la relacionada con la medida del rendimiento del servidor de bases de datos.

Esta utilidad se puede descargar desde el sitio <http://sf.net/projects/sysbench/>. Este programa se distribuye libremente bajo licencia GNU. En la dirección Web anterior esta disponible también el código fuente de la aplicación. Para utilizarlo, solamente se tendrá que compilar e instalar. Una vez hecho esto el programa estará disponible para su uso. La sintaxis general para ejecutar Sysbench es la siguiente:

```
sysbench [opciones_comunes] --test=nombre_test [opciones_test] comando
```

En la sintaxis anterior, *opciones_comunes* hace referencia a las opciones del programa independientes del test que se va a realizar, *nombre_test* es el nombre de la prueba que se quiere ejecutar, *opciones_test* son las opciones para esa prueba en concreto y *comando* describe la acción que Sysbench debe hacer. Los comandos disponibles son los siguientes:

- *prepare*. Realiza acciones preparativas para aquellos tests que las necesiten, por ejemplo, llenando con datos las bases de datos que se van a usar para el test *oltp*.
- *run*. Ejecuta un test determinado.
- *cleanup*. Borra los datos temporales creados por un determinado test, en el caso de que se hayan necesitado.
- *help*. Muestra información sobre un test determinado en pantalla.

Las opciones comunes que se pueden utilizar, así como una breve descripción de las mismas se muestran en la tabla 6.18.

Opción	Descripción
--num-threads	Número de total de hilos (usuarios) a crear.
--max-requests	Límite máximo de solicitudes.
--max-time	Tiempo máximo de ejecución del test (0 significa ilimitado).
--thread-stack-size	Tamaño de la pila para cada hilo.
--test	Nombre del test que se va a ejecutar.
--debug	Se muestra información adicional para depuración.
--validate	Ejecuta una validación de los resultados del test en el caso de que sea posible.
--verbosity	Nivel de detalle de la información de depuración (0 - sólo mensajes críticos, 5 - <i>debug</i>).
--percentile	Sysbench mide los tiempos de ejecución de todas las tareas implementadas para mostrar información estadística como mínimos, promedios, máximos y percentiles. Esta opción permite especificar el percentil a utilizar (por defecto el 95%).

Tabla 6.18. Opciones comunes de Sysbench.

De todas las pruebas que se pueden realizar con esta aplicación, el que realmente interesa para el propósito de este Proyecto Fin de Carrera es el test *oltp*, que permite evaluar el rendimiento de un sistema gestor de bases de datos. Para ejecutar este test, primero se deberá ejecutar Sysbench con el comando *prepare*. Esto creará por defecto las tablas que se van a utilizar para las pruebas. El test *oltp* se puede efectuar de tres formas:



- Simple. En este modo cada usuario ejecuta una consulta SELECT simple.
- Transaccional avanzado. Cada usuario realiza transacciones sobre las tablas del test. Si estas tablas soportan transacciones (p.ej. las almacenadas con el motor InnoDB en MySQL o las de PostgreSQL), se usarán las instrucciones BEGIN/COMMIT para comenzar/parar la transacción. Si la tabla no soporta control transaccional (por ejemplo las almacenadas con el motor MyISAM en MySQL) se usarán las instrucciones LOCK TABLES/UNLOCK TABLES de la forma que se explicó en un apartado anterior cuando se discutió la forma en que MySQL puede superar la carencia del control transaccional. Cada transacción comprende varias instrucciones SQL.
- No transaccional. Este modo es similar al modo simple, pero en este caso se puede elegir la consulta que se va a ejecutar.

Para las pruebas que se realizarán en este proyecto se utilizará el modo Transaccional avanzado.

A continuación, en la tabla 6.19 se muestra un listado de las principales opciones específicas del test *oltp* que se pueden utilizar:

Opción	Descripción
--oltp-test-mode	Especifica el modo de ejecución del test. Los posibles valores son: <i>simple</i> (simple), <i>complex</i> (transaccional avanzado) y <i>nontrx</i> (no transaccional). Por defecto se ejecuta el test transaccional avanzado.
--oltp-read-only	Se ejecutan instrucciones de solo lectura (no se emplean sentencias UPDATE, DELETE o INSERT).
--oltp-range-size	Tamaño de las consultas.
--oltp-point-selects	Número de consultas SELECT en una transacción simple.
--oltp-index-updates	Número de consultas UPDATE en una tabla indexada en una misma transacción.
--oltp-non-index-updates	Número de consultas UPDATE en una tabla no indexada en una misma transacción.
--oltp-nontrx-mode	Tipo de consultas para el modo de ejecución no transaccional. Posibles valores: <i>select</i> , <i>update_key</i> , <i>update_nokey</i> , <i>insert</i> y <i>delete</i> .
--oltp-connect-delay	Tiempo en milisegundos entre cada conexión a la base de datos.
--oltp-user-delay-min	Tiempo mínimo de espera después de una consulta, en milisegundos.
--oltp-user-delay-max	Tiempo máximo de espera después de una consulta, en milisegundos.
--oltp-table-name	Nombre de la tabla que se va a emplear para las pruebas. Por defecto se utiliza la tabla <i>sbtest</i> .
--oltp-table-size	Número de filas de la tabla que se va a utilizar en las pruebas. Por defecto 10000.

Tabla 6.19. Opciones del test OLTP.

Además de estas opciones, para cada gestor de bases de datos existen unas opciones adicionales, que se muestran en las tablas 6.20 y 6.21.

Opción	Descripción
--mysql-host	Nombre del host donde se encuentra el servidor MySQL.
--mysql-port	Número de puerto del servidor MySQL.
--mysql-socket	Dirección donde se encuentra el <i>socket</i> UNIX para la comunicación con la base de datos.
--mysql-user	Nombre del usuario MySQL.
--mysql-password	Clave de usuario MySQL.



--mysql-db	Nombre de la base de datos que se va a emplear en los test. Por defecto se toma <i>sbtest</i> . Nótese que Sysbench no crea esta base de datos, sólo las tablas. Por tanto, se debe crear esta base de datos y darle los privilegios adecuados al usuario que hará los test antes de ejecutar los mismos.
--mysql-table-type	Motor de almacenamiento empleado por las tablas. Posibles valores: <i>myisam</i> , <i>innodb</i> , <i>heap</i> , <i>ndbcluster</i> , <i>bdb</i> .
--myisam-max-rows	Número máximo de filas para tablas MyISAM. Por defecto 1000000.

Tabla 6.20. Opciones de la prueba OLTP para MySQL.

Opción	Descripción
--pgsql-host	Nombre del host donde se encuentra el servidor PostgreSQL.
--pgsql-port	Número de puerto del servidor PostgreSQL.
--pgsql-user	Nombre del usuario PostgreSQL.
--pgsql-password	Clave de usuario.
--pgsql-db	Nombre de la base de datos que se va a emplear en los test. Por defecto se toma <i>sbtest</i> . Nótese que Sysbench no crea esta base de datos, sólo las tablas. Por tanto, se debe crear esta base de datos y darle los privilegios adecuados al usuario que hará los test antes de ejecutar los mismos.

Tabla 6.21. Opciones de la prueba OLTP para PostgreSQL.

Una vez que concluya el test, se mostrarán por pantalla los resultados obtenidos del mismo. Se muestran el número de consultas efectuadas, el número total de transacciones, el número de transacciones por segundo, el número de bloqueos ocurridos, el número de operaciones de lectura/escritura realizadas, el número de operaciones de lectura/escritura por segundo, el tiempo total empleado por el test, tiempos máximos, mínimos, promedio y percentil por solicitud, y alguna información adicional mas.

Este test se ha realizado para ver el rendimiento de los servidores de bases de datos a estudio para distinta carga de usuarios.

6.4.2. Resultados obtenidos.

Para comparar los resultados de este banco de pruebas para cada uno de los sistemas gestores de bases de datos a estudio, se ha diseñado una batería de pruebas con el fin de determinar cómo afecta la carga de usuarios al rendimiento del sistema. Las pruebas se han realizado varias veces para 1, 5, 30, 60, 90, 150, 225 y 300 usuarios. En el ANEXO B se mostrará el listado completo obtenido para cada una de estas pruebas. De estos listados, se han extraído los datos mas importantes, que se van a mostrar a continuación en forma de gráfica. En estas gráficas se van a mostrar el número de transacciones por segundo, las operaciones de lectura realizadas por segundo, el número de operaciones de escritura realizadas por segundo y el tiempo promedio de ejecución de consulta, todo ello en función del número de usuarios. Dichas gráficas se muestran en las figuras 6.26, 6.27, 6.28 y 6.29.

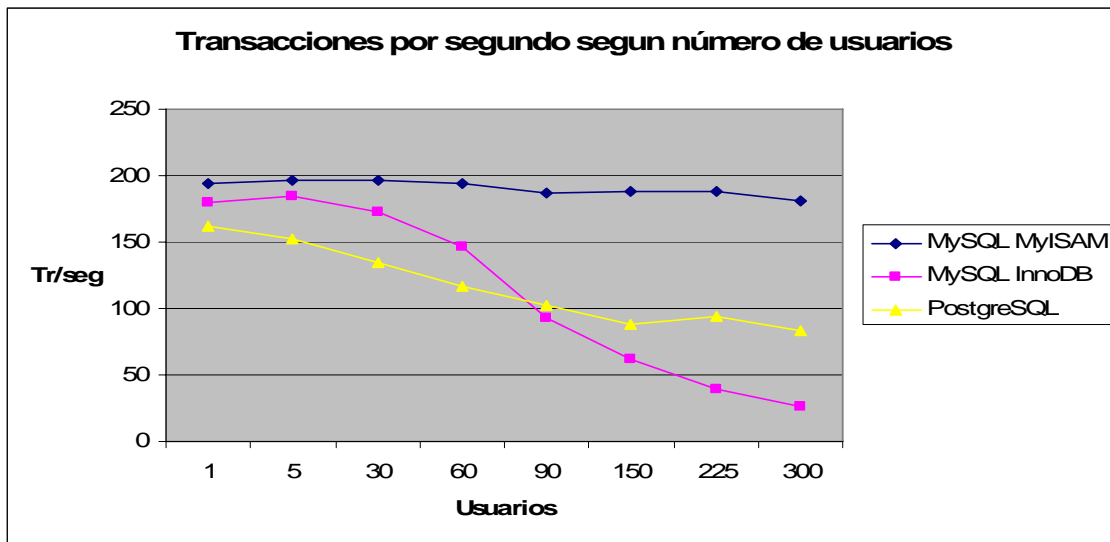


Figura 6.26. Número de transacciones por segundo en función del número de usuarios.

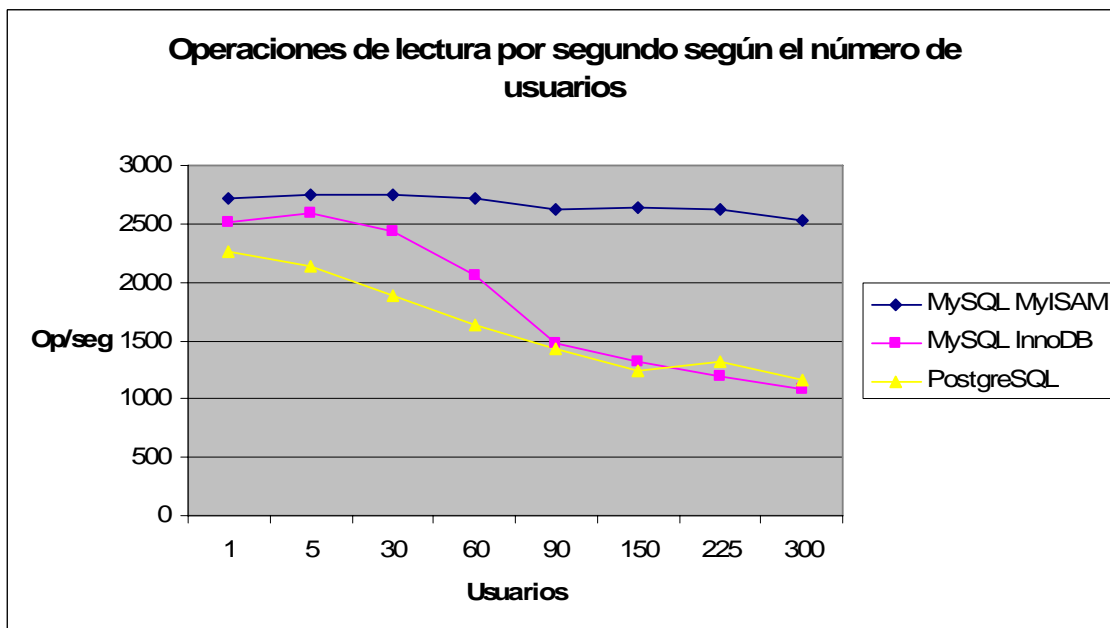


Figura 6.27. Número de operaciones de lectura por segundo en función del número de usuarios.

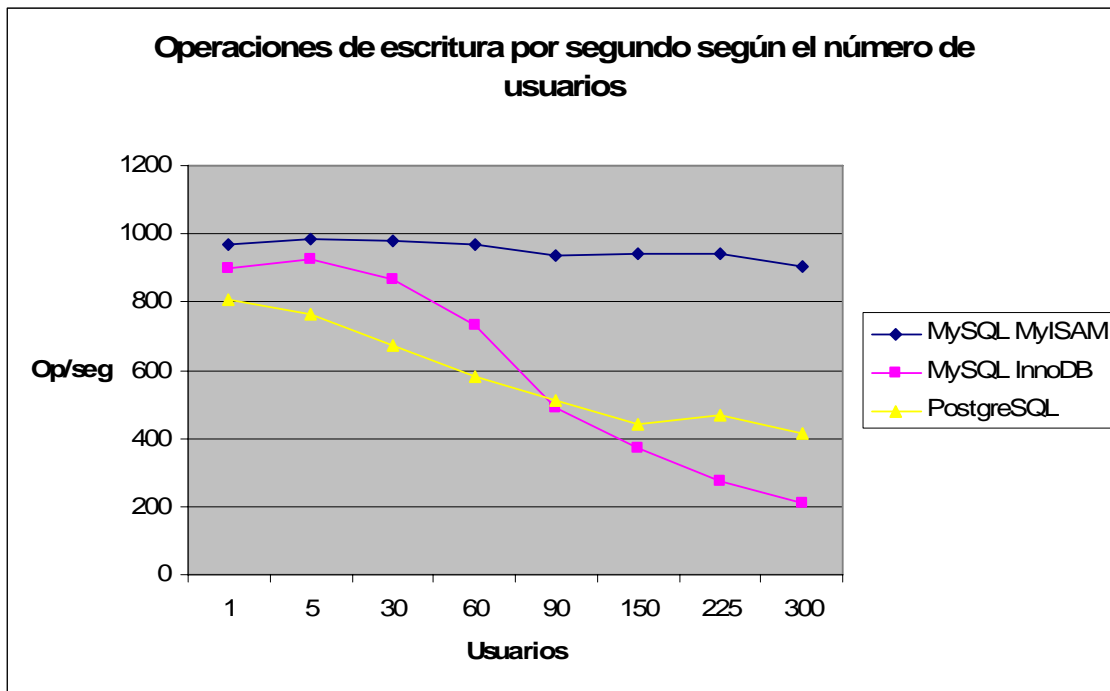


Figura 6.28. Número de operaciones de escritura por segundo en función del número de usuarios.

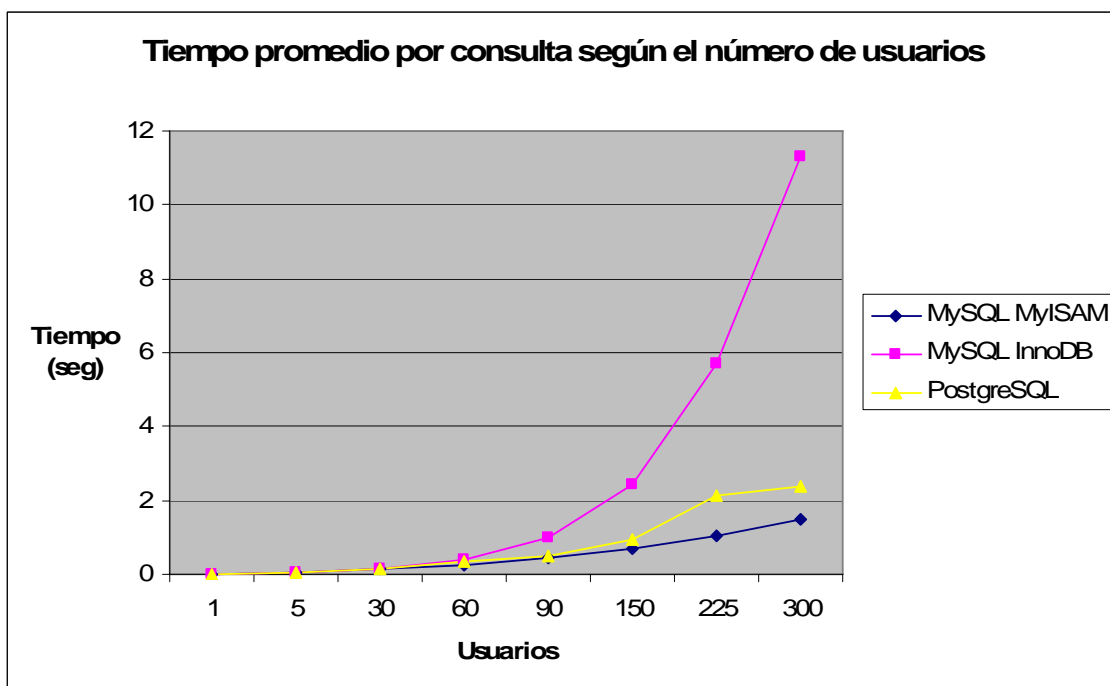


Figura 6.27. Tiempo promedio por consulta en función del número de usuarios.

6.4.3. Comentarios.

A la vista de las gráficas anteriores, se pueden obtener diversas conclusiones.

Si se observa la gráfica dedicada a las transacciones, se observa que para MySQL con motor de almacenamiento MyISAM, el número de transacciones apenas decrece cuando



aumenta la carga de usuarios (dentro del límite de usuarios de las pruebas), si bien es cierto que estas transacciones no son implementadas como tal, sino mediante el uso de bloqueos. Si se observan los otros dos casos, se puede ver que para una cantidad de usuarios inferior a 100 (aproximadamente), MySQL con el motor de almacenamiento InnoDB ejecuta un mayor número de transacciones por segundo que PostgreSQL. Sin embargo, cuando la carga de usuarios aumenta, PostgreSQL tiene mayor rendimiento que MySQL con el motor de almacenamiento InnoDB. También hay que comentar que tanto PostgreSQL como MySQL con motor de almacenamiento InnoDB ven notablemente empeorado su rendimiento a medida que aumenta el número de usuarios, como muestra la elevada tendencia decreciente de ambas líneas.

Conclusiones parecidas se pueden obtener a partir de las gráficas correspondientes al número de operaciones de lectura y escritura por segundo. En dichas gráficas, se ve como MySQL con motor de almacenamiento MyISAM ofrece el mayor número de operaciones, y dicho número apenas decrece cuando aumenta la carga de usuarios. En el extremo opuesto están los otros dos casos, en los que se ve que cuando el número de usuarios aumenta, disminuyen drásticamente el número de operaciones de lectura o escritura realizadas por segundo.

En cuanto al tiempo promedio por consulta según el número de usuarios, se puede ver que cuando el número de usuarios no es muy elevado, dicho promedio es parecido en los tres casos, siendo menor para MySQL con motor de almacenamiento MyISAM. La diferencia significativa se aprecia cuando se aumenta el número de usuarios a una cantidad superior a cien, en cuyo caso el tiempo promedio para PostgreSQL aumenta sensiblemente respecto al correspondiente a MySQL con MyISAM. Pero la diferencia más acusada se ve en el caso de MySQL con motor InnoDB, en el cual este tiempo promedio aumenta de manera importante según el número de usuarios aumenta.

Por último, cabe comentar que el número máximo de usuarios simultáneos que se ha empleado en las pruebas, cuyo valor es 300, viene dado por los recursos físicos disponibles para la realización de las pruebas. Cuando se intentó ejecutar Sysbench para valores de número de usuarios simultáneos superiores a 300, el rendimiento del sistema caía notablemente, detectándose en bastantes ocasiones situaciones de inestabilidad en los sistemas. Si se deseara extender a esta prueba para más de 300 usuarios simultáneos se debería disponer de un equipo con más memoria RAM (recuerde que la cantidad de memoria que se ha utilizado para la ejecución de estas pruebas es de 1 GB).