

Optimización y rendimiento en MySQL



IVANA SÁNCHEZ PÉREZ-2ºASIR

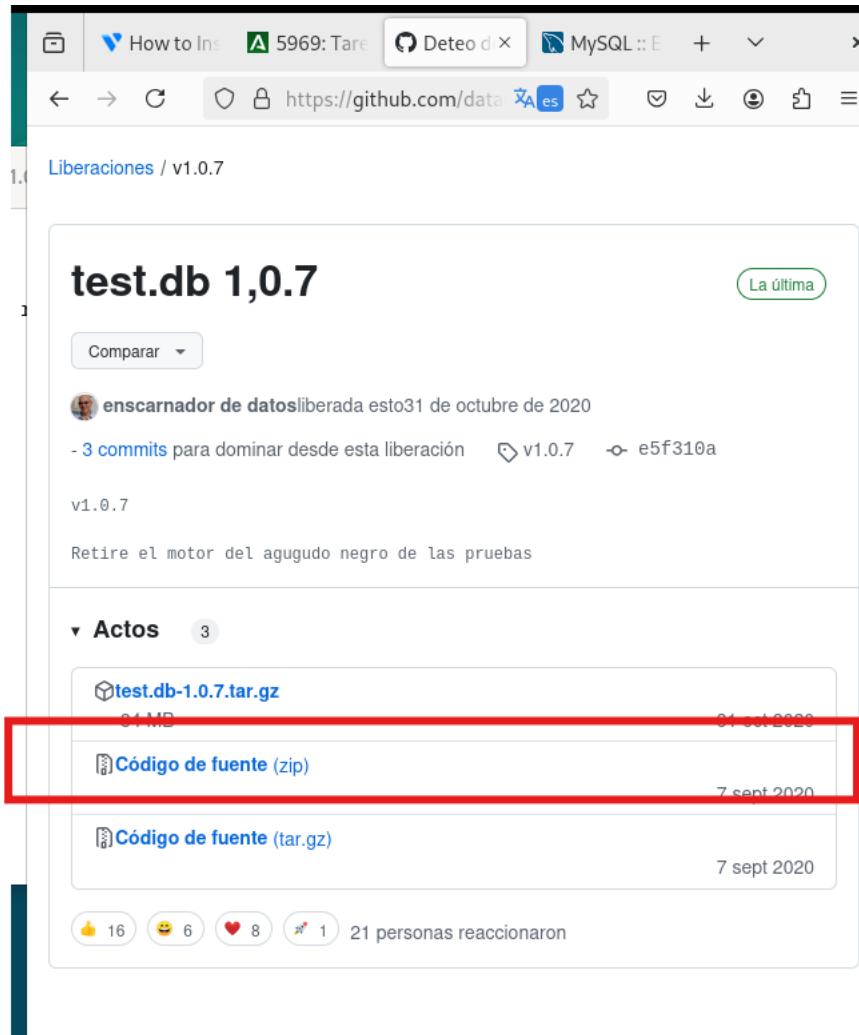
MySQL

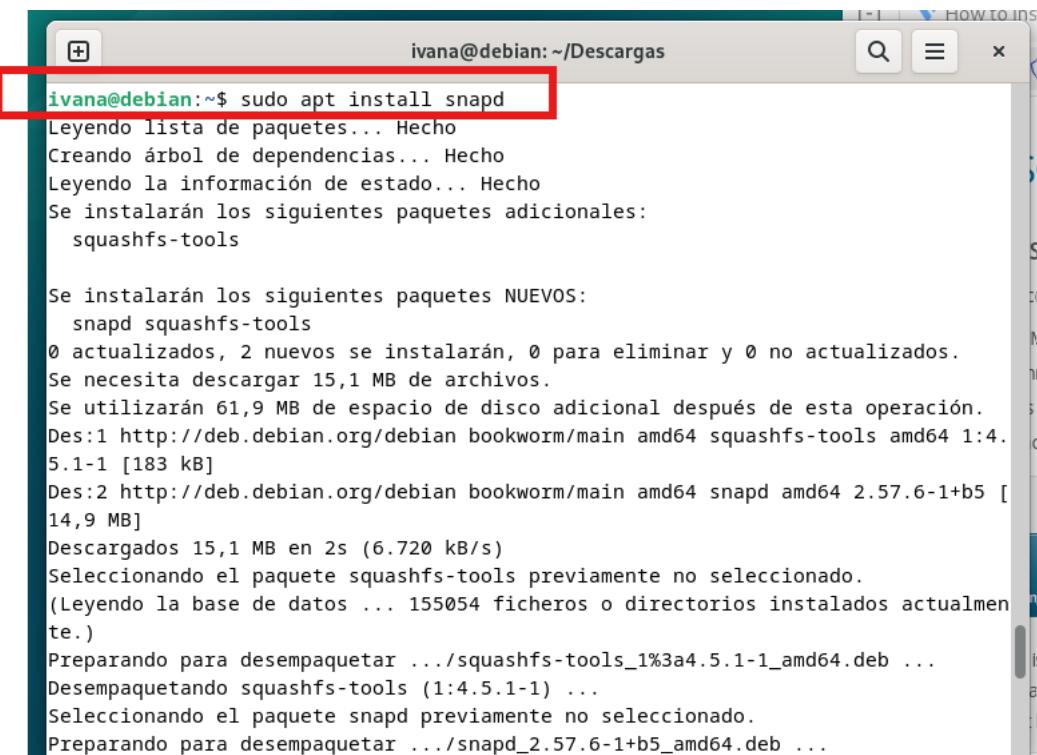
1.- Introducción

En esta tarea tiene como objetivo exemplificar de forma práctica algunos de los aspectos más relevantes para que una base de dato se mantenga en marcha en las mejores condiciones de rendimiento.

2.- Apartado 1

Voy a utilizar una máquina virtual Debian12 para realizar la tarea. Para ello, lo primero que voy a hacer es descargar MySQL a través de su terminal, con sus correspondientes archivos debug, y el MySQL Workbench, para trabajar de una forma más cómoda con una interfaz gráfica.



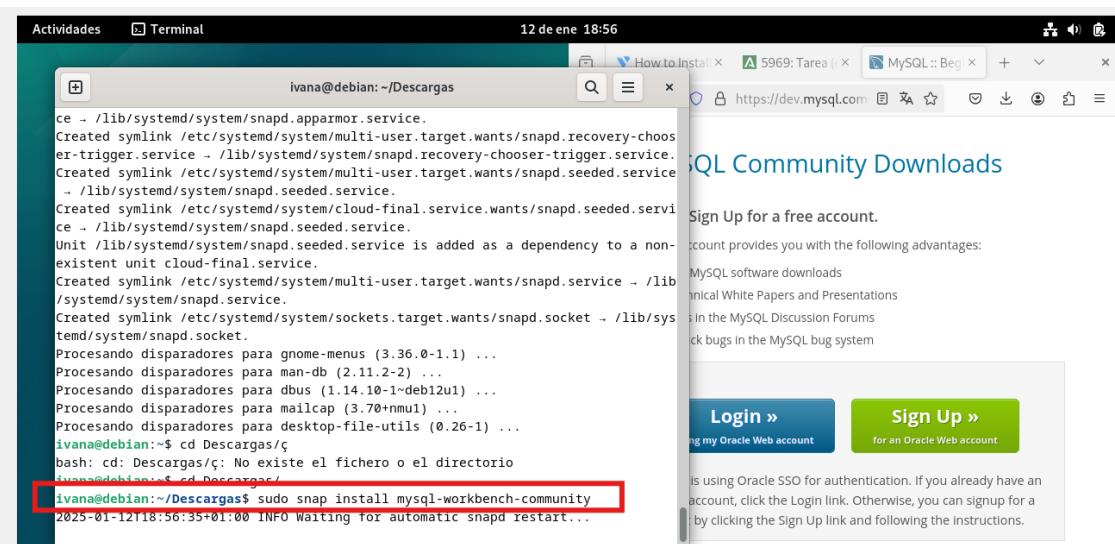


```

ivana@debian:~/Descargas$ sudo apt install snapd
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  squashfs-tools

Se instalarán los siguientes paquetes NUEVOS:
  snapd squashfs-tools
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 15,1 MB de archivos.
Se utilizarán 61,9 MB de espacio de disco adicional después de esta operación.
Des:1 http://deb.debian.org/debian bookworm/main amd64 squashfs-tools amd64 1:4.5.1-1 [183 kB]
Des:2 http://deb.debian.org/debian bookworm/main amd64 snapd amd64 2.57.6-1+b5 [14,9 MB]
Descargados 15,1 MB en 2s (6.720 kB/s)
Seleccionando el paquete squashfs-tools previamente no seleccionado.
(Leyendo la base de datos ... 155054 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../squashfs-tools_1%3a4.5.1-1_amd64.deb ...
Desempaquetando squashfs-tools (1:4.5.1-1) ...
Seleccionando el paquete snapd previamente no seleccionado.
Preparando para desempaquetar .../snapd_2.57.6-1+b5_amd64.deb ...

```



```

ivana@debian:~/Descargas$ sudo snap install mysql-workbench-community
2025-01-12T18:56:35+01:00 INFO Waiting for automatic snapd restart...

```

Una vez instalados, procedemos a descargarnos la base de datos de prueba que MySQL ha puesto a disposición en su web, llamada `employees.sql`, y que descargaremos en el terminal y comprobaremos que se ha creado a la misma vez en Workbench.

Actividades Terminal 12 de ene 18:30

```

-> e^C
mysql> exit
Bye
ivana@debian:~/Descargas/test_db-1.0.7$ sudo mysql -u root -p < employees.sql
Enter password:
INFO
CREATING DATABASE STRUCTURE
INFO
storage engine: InnoDB
INFO
LOADING departments
INFO
LOADING employees
INFO
LOADING dept_emp
INFO
LOADING dept_manager
INFO
LOADING titles
INFO
LOADING salaries
data_load_time_diff
00:03:26
ivana@debian:~/Descargas/test_db-1.0.7$ 

```

Popular open source database Contact MySQL

from Employees DB on GitHub. You can download a access the information through Git.

load the archive and unpack it using WinZip or another ange location into the unpacked package directory. For commands:

ople with several different storage engines, with the InnoDB Employees.sql file and adjust the comments to choose a

Actividades MySQL Workbench 12 de ene 20:20

MySQL Workbench

Schemas: employees

Query1: departments

Result Grid:

#	dept_no	dept_name
1	d009	Customer Service
2	d005	Development
3	d002	Finance
4	d003	Human Resources
5	d001	Marketing
6	d004	Production
7	d006	Quality Management
8	d008	Research
9	d007	Sales

Object Info Session departments 1 Apply

Query Completed

3.- Apartado 2

En este apartado trataremos la optimización del espacio en disco, y para ello empezaremos creando una nueva tabla llamada employeesCHAR.

CHAR y VACHAR son tipos de datos que pueden almacenar cadenas de caracteres de diferentes longitudes. La diferencia entre ambas estriba en la forma en que asignan y manejan el espacio para las cadenas. CHAR significa cadena de caracteres de longitud fija, mientras que VARCHAR significa cadena de caracteres de longitud variable.

The screenshot shows the MySQL Workbench interface with the 'employees' schema selected. A new table named 'employeesCHAR' is being created under the 'employees' schema. The table structure includes columns for emp_no (INT), birth_date (DATE), first_name (CHAR(14)), last_name (CHAR(16)), and gender (ENUM('M', 'F')). The 'Generated Column Storage Type' is set to 'VIRTUAL'. The 'Indexes' tab is currently selected.

The screenshot shows the MySQL Workbench interface with the 'employees' schema selected. An SQL script is being run in the Query Editor:

```
1 • USE employees;
2 • INSERT INTO employeesCHAR SELECT * FROM employees;
```

MySQL Workbench

Local instance 3306

File Edit View Query Database Server Tools Scripting Help

Administration Schemas

SCHEMAS

employees

- Tables
 - departments
 - dept_emp
 - dept_manager
 - employees
 - employeesCHAR
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - salaries
 - titles
- Views
- Stored Procedures
- Functions

Object Info Session

Query Completed

Result Grid

```
1 • SELECT * FROM employees.employeesCHAR;
```

#	emp_no	birth_date	first_name	last_name	gender	hire_date
1	10001	1953-09-02	Georgi	Facello	M	1986-06-26
2	10002	1964-06-02	Bezalel	Simmel	F	1985-11-21
3	10003	1959-12-03	Parto	Bamford	M	1986-08-28
4	10004	1954-05-01	Christian	Koblick	M	1986-12-01
5	10005	1955-01-01	Khalid	M	M	1986-08-12

employeesCHAR1

Se observará que hay una diferencia de aproximadamente 5MB entre las dos tablas.

Actividades MySQL Workbench 12 de ene 20:43

Local instance 3306

Edit View Query Database Server Tools Scripting Help

Administration Schemas

EMAS

employees

- Tables
 - departments
 - dept_emp
 - dept_manager
 - employees
 - employeesCHAR
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - salaries
 - titles
- Views
- Stored Procedures
- Functions
- sys

Object Info Session

Query Completed

```
1
2 • SELECT table_name AS "Tabla",
3     ROUND ((data_length + index_length) / 1024 / 1024), 2) AS "Tamaño (MB)"
4     FROM information_schema.TABLES
5     WHERE table_schema = "employees"
6     ORDER BY (data_length + index_length) DESC;
7
```

Result Grid

#	Tabla	Tamaño (MB)
2	titles	19.56
3	employeesCHAR	19.55
4	dept_emp	17.03
5	employees	14.52
6	departments	0.03

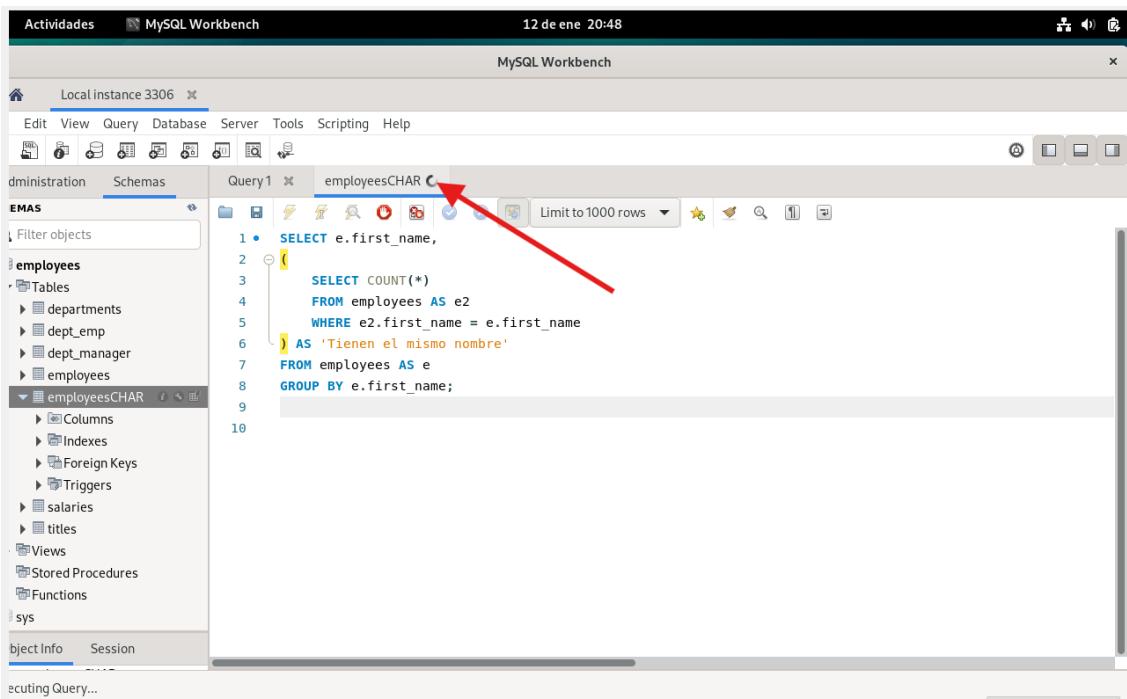
Result 2

Read Only

4.- APARTADO 3

Apartado que trata de la optimización de consultas a través de la sentencia EXPLAIN.

Primero hacemos una consulta normal pidiendo un listado con los nombres de los empleados y la cantidad de ellos que tienen el mismo nombre (Consulta pesada).

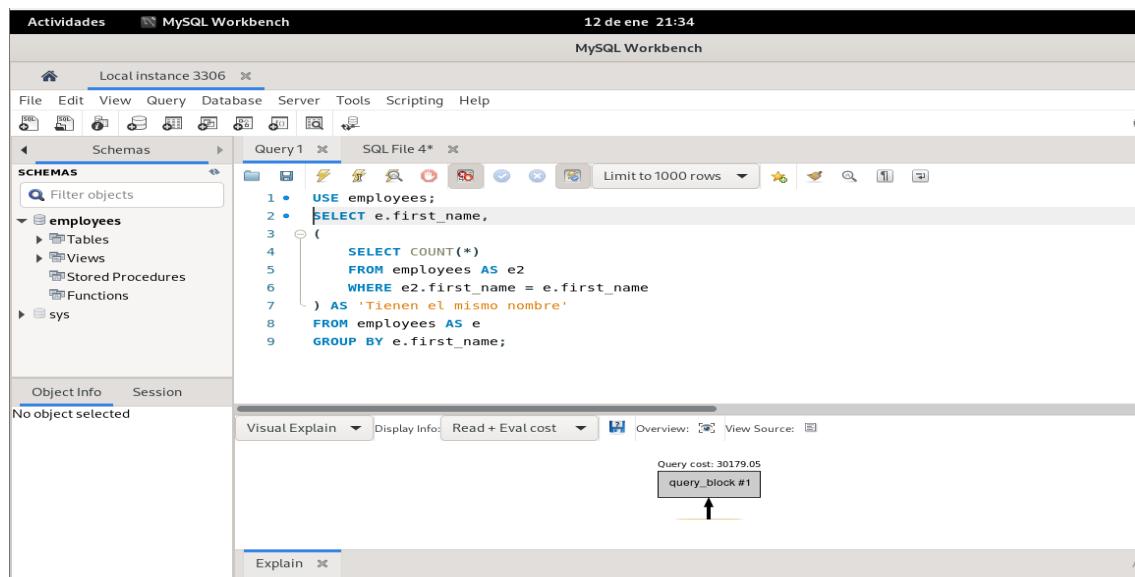


The screenshot shows the MySQL Workbench interface. In the left sidebar, under the 'Schemas' section, the 'employeesCHAR' schema is selected. In the main area, a query named 'Query 1' is displayed in the 'Query Editor' tab:

```
1 • SELECT e.first_name,
2   (
3     SELECT COUNT(*)
4       FROM employees AS e2
5         WHERE e2.first_name = e.first_name
6   ) AS 'Tienen el mismo nombre'
7   FROM employees AS e
8   GROUP BY e.first_name;
```

A red arrow points from the text above the screenshot to the 'Execute' button in the toolbar of the Query Editor.

El tiempo de ejecución se dispara a los 1000 segundos que tenemos por defecto como límite. El resultado final es un error.



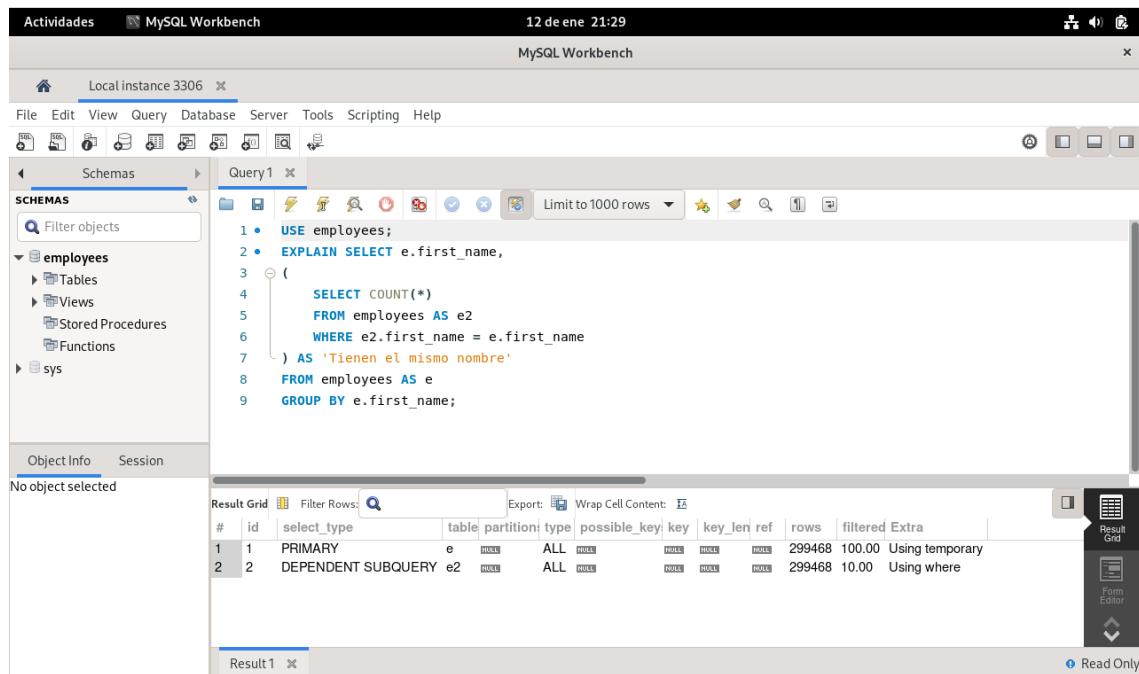
The screenshot shows the MySQL Workbench interface with the 'Explain' tab selected. The same query is shown, but the execution has been forced to run by setting the 'Limit to 1000 rows' dropdown to '0'. The results show a very high query cost of 30179.05, indicating a performance issue:

```
1 • USE employees;
2 • SELECT e.first_name,
3   (
4     SELECT COUNT(*)
5       FROM employees AS e2
6         WHERE e2.first_name = e.first_name
7   ) AS 'Tienen el mismo nombre'
8   FROM employees AS e
9   GROUP BY e.first_name;
```

The 'Explain' tab displays the following information:

- Visual Explain
- Display Info
- Read + Eval cost: 30179.05
- query_block #1

Aplicaremos la sentencia EXPLAIN delante de SELECT y observamos que no sólo no da error, sino que da un resultado inmediato.



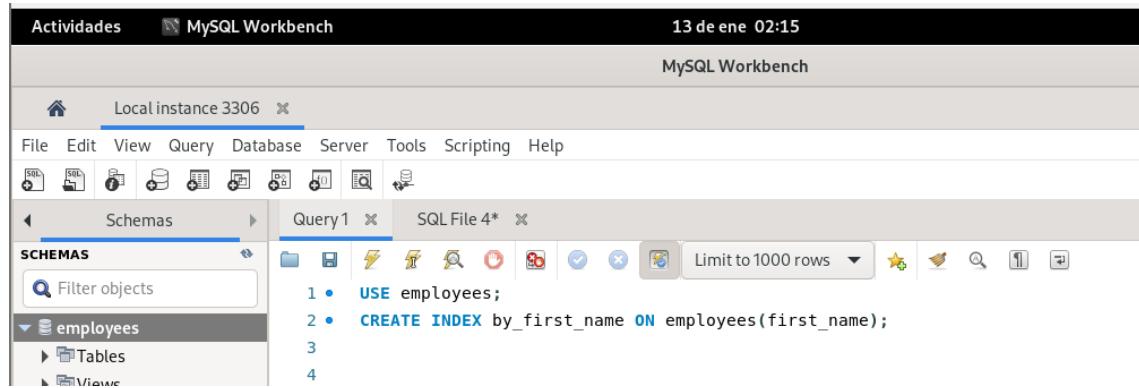
The screenshot shows the MySQL Workbench interface. In the left sidebar, under 'SCHEMAS', the 'employees' schema is selected. In the main pane, a query window titled 'Query 1' contains the following SQL code:

```
1 • USE employees;
2 • EXPLAIN SELECT e.first_name,
3   (
4     SELECT COUNT(*)
5     FROM employees AS e2
6     WHERE e2.first_name = e.first_name
7   ) AS 'Tienen el mismo nombre'
8   FROM employees AS e
9   GROUP BY e.first_name;
```

Below the query window, the 'Result Grid' shows the execution plan:

#	id	select_type	table	partition	type	possible_key	key	key_len	ref	rows	filtered	Extra
1	1	PRIMARY	e	HULL	ALL	HULL	HULL	HULL	HULL	299468	100.00	Using temporary
2	2	DEPENDENT SUBQUERY	e2	HULL	ALL	HULL	HULL	HULL	HULL	299468	10.00	Using where

Ahora vamos a crear un índice y ejecutaremos una consulta. Esto hará que ésta se ejecute en menos de 1 segundo.



The screenshot shows the MySQL Workbench interface. In the left sidebar, under 'SCHEMAS', the 'employees' schema is selected. In the main pane, a query window titled 'Query 1' contains the following SQL code:

```
1 • USE employees;
2 • CREATE INDEX by_first_name ON employees(first_name);
```

Actividades MySQL Workbench 13 de ene 02:57

MySQL Workbench Local instance 3306

File Edit View Query Database Server Tools Scripting Help

Schemas departments employees

Filter objects

employees

Tables

departments dept_emp dept_manager employees

Columns Indexes Foreign Keys

Object Info Session

Connection Details

Name: Local instance 3306

Host: localhost Port: 3306

Login User: root Current User: root@localhost

SSL cipher: SSL not used

Server

Query 1 departments employees

1 • SHOW INDEX FROM employees;

Result Grid Filter Rows Export Wrap Cell Content

#	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed
1	employees	0	PRIMARY	1	emp_no	A	299468	NULL	NULL
2	employees	1	by_first_name	1	first_name	A	1282	NULL	NULL

Result 7

The screenshot shows the MySQL Workbench interface with the 'employees' schema selected. In the 'Query' tab, the command 'SHOW INDEX FROM employees;' is run. The results are displayed in a grid. The first row, representing the primary key, has a red box around it. The second row, representing an index named 'by_first_name' on the 'first_name' column, is also highlighted with a red box.

Hacemos una consulta con el índice.

Actividades MySQL Workbench 13 de ene 02:59

Local instance 3306

File Edit View Query Database Server Tools Scripting Help

Schemas Query 1 departments employees

SCHEMAS

employees

- Tables
 - departments
 - dept_emp
 - dept_manager
 - employees
 - Columns
 - Indexes
 - Foreign Keys

Object Info Session

Connection Details
Name: Local instance
3306
Host: localhost
Port: 3306
Login User: root
Current User: root@localhost
SSL cipher: SSL not used
Server

```

1 • SELECT e.first_name,
2   (
3     SELECT COUNT(*)
4       FROM employees AS e2
5         WHERE e2.first_name = e.first_name
6   ) AS 'Tienen el mismo nombre'
7   FROM employees AS e
8     GROUP BY e.first_name;
9
10
11
12

```

Result Grid Filter Rows: Export: Wrap Cell Content: Fetch rows:

#	first_name	Tienen el mismo nombre
1	Aamer	228
2	Aamod	216
3	Abdelaziz	227
4	Abdelghani	247
5	Abdelkader	222
6	Abdelwaheb	241

Result 1

Query Completed

CTRL DERECHA

Y una segunda consulta con EXPLAIN

Actividades MySQL Workbench 13 de ene 03:01

Local instance 3306

File Edit View Query Database Server Tools Scripting Help

Schemas Query 1 departments employees

SCHEMAS

employees

- Tables
 - departments
 - dept_emp
 - dept_manager
 - employees
 - Columns
 - Indexes
 - Foreign Keys

Object Info Session

Connection Details
Name: Local instance
3306
Host: localhost
Port: 3306
Login User: root
Current User: root@localhost
SSL cipher: SSL not used
Server

```

1 • EXPLAIN SELECT e.first_name,
2   (
3     SELECT COUNT(*)
4       FROM employees AS e2
5         WHERE e2.first_name = e.first_name
6   ) AS 'Tienen el mismo nombre'
7   FROM employees AS e
8     GROUP BY e.first_name;
9
10
11
12

```

Result Grid Filter Rows: Export: Wrap Cell Content: rows filtered Extra

#	id	select_type	table	partition	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	1	PRIMARY	e		range	by_first_name	by_first_name	58		1283	100.00	Using index for group-by
2	2	DEPENDENT SUBQUERY	e2		ref	by_first_name	by_first_name	58	func	233	100.00	Using index

Result 2

Read Only

5.- Apartado 4

Fichero log de consultas lentas, es decir, fichero en el que se registrarán las consultas que hayan sobrepasado el límite de tiempo establecido.

Procedo a averiguar si tengo activado dicho fichero, y su ruta.

The screenshot shows the MySQL Workbench interface. In the top right, it says "13 de ene 03:09". The main area has a query editor window titled "Query 1" containing the following SQL command:

```
SHOW VARIABLES LIKE '%slow_query_log%';
```

The result grid shows two variables:

#	Variable_name	Value
1	slow_query_log	ON
2	slow_query_log_file	/var/lib/mysql/debian-slow.log

Below the query editor, there is a "Connection Details" panel for "Local instance 3306" with the following information:

- Name: Local instance
- Host: localhost
- Port: 3306
- Login User: root

At the bottom, there is a terminal window titled "ivana@debian: ~" showing the output of the command:

```
root@debian:/home/ivana# cat /var/lib/mysql/debian-slow.log
/usr/sbin/mysqld, Version: 8.0.40 (MySQL Community Server - GPL). started with:
Tcp port: 3306 Unix socket: /var/run/mysqld/mysqld.sock
Time          Id Command    Argument
root@debian:/home/ivana#
```

```
ivana@debian:~  
# Query_time: 0.000768 Lock_time: 0.000007 Rows_sent: 0 Rows_examined: 0  
SET timestamp=1736768302;  
SELECT st.* FROM performance_schema.events_stages_history_long st WHERE st.nesting_event_id = 54;  
# Time: 2025-01-13T11:38:22.441999Z  
# User@Host: root[root] @ localhost [::1] Id: 9  
# Query_time: 0.000967 Lock_time: 0.000005 Rows_sent: 0 Rows_examined: 0  
SET timestamp=1736768302;  
SELECT st.* FROM performance_schema.events_waits_history_long st WHERE st.nesting_event_id = 54;  
# Time: 2025-01-13T11:38:32.126766Z  
# User@Host: root[root] @ localhost [::1] Id: 10  
# Query_time: 3.877829 Lock_time: 0.000009 Rows_sent: 1000 Rows_examined: 2844047  
use employees;  
SET timestamp=1736768308;  
SELECT s.salary,  
      COUNT(*) AS 'Número de empleados con este título'  
FROM salaries AS s  
GROUP BY s.salary  
LIMIT 0, 1000;  
# Time: 2025-01-13T11:42:46.925057Z  
# User@Host: root[root] @ localhost [::1] Id: 9  
# Query_time: 0.000262 Lock_time: 0.000002 Rows_sent: 0 Rows_examined: 0  
SET timestamp=1736768566;  
SELECT st.* FROM performance_schema.events_stages_history_long st WHERE st.nesting_event_id = 71;  
# Time: 2025-01-13T11:42:46.926216Z  
# User@Host: root[root] @ localhost [::1] Id: 9  
# Query_time: 0.000235 Lock_time: 0.000001 Rows_sent: 0 Rows_examined: 0  
SET timestamp=1736768566;  
SELECT st.* FROM performance_schema.events_waits_history_long st WHERE st.nesting_event_id = 71;  
root@debian:/home/ivana#
```

6.- Apartado 5

Para este apartado voy a centrarme en la tabla salaries para obtener el salario promedio de los empleados de cada departamento.

a) Consulta inicial

Debian [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Actividades MySQL Workbench 13 de ene 15:01

MySQL Workbench

Local instance 3306

File Edit View Query Database Server Tools Scripting Help

Schemas Query 1 SQL File 5*

Filter objects

1 • USE employees;

2

3 • SELECT d.dept_name, AVG(s.salary) AS avg_salary

4 FROM departments AS d

5 JOIN dept_emp AS de ON d.dept_no = de.dept_no

6 JOIN salaries AS s ON de.emp_no = s.emp_no

7 GROUP BY d.dept_name;

8

Object Info Session

Connection Details

Name: Local instance 306

Host: localhost

Port: 3306

Login User: root

Current User: root@localhost

SSL cipher: SSL not used

Result Grid

Filter Rows: Export: Wrap Cell Content: rows filter

#	id	select_type	table	partition	type	possible_keys	key	key_len	ref	rows	filter
1	1	SIMPLE	de		index	PRIMARY,dept_no	dept_no	16	NULL	33143	100
2	1	SIMPLE	d		eq_ref	PRIMARY,dept_name	PRIMARY	16	employees.de.dept_no	1	100
3	1	SIMPLE	s		ref	PRIMARY	PRIMARY	4	employees.de.emp_no	9	100

b) Consulta con el comando EXPLAIN

Debian [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Actividades MySQL Workbench 13 de ene 15:02

MySQL Workbench

Local instance 3306

File Edit View Query Database Server Tools Scripting Help

Schemas Query 1 SQL File 5*

Filter objects

1 • USE employees;

2

3 • EXPLAIN SELECT d.dept_name, AVG(s.salary) AS avg_salary

4 FROM departments AS d

5 JOIN dept_emp AS de ON d.dept_no = de.dept_no

6 JOIN salaries AS s ON de.emp_no = s.emp_no

7 GROUP BY d.dept_name;

8

Object Info Session

Connection Details

Name: Local instance 306

Host: localhost

Port: 3306

Login User: root

Current User: root@localhost

SSL cipher: SSL not used

Result Grid

Filter Rows: Export: Wrap Cell Content: rows filter

#	id	select_type	table	partition	type	possible_keys	key	key_len	ref	rows	filter
1	1	SIMPLE	de		index	PRIMARY,dept_no	dept_no	16	NULL	33143	100
2	1	SIMPLE	d		eq_ref	PRIMARY,dept_name	PRIMARY	16	employees.de.dept_no	1	100
3	1	SIMPLE	s		ref	PRIMARY	PRIMARY	4	employees.de.emp_no	9	100

c) Creación de un índice

The screenshot shows two instances of MySQL Workbench running on a Debian system.

Top Window (Query 1):

```
1 • USE employees;
2 • CREATE INDEX idx_emp_no ON salaries(emp_no);
```

Bottom Window (Query 6):

```
1 • show index FROM salaries;
```

Result Grid:

#	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
1	salaries	0	PRIMARY	1	emp_no	A	294479	NULL	NULL	NULL	BTREE
2	salaries	0	PRIMARY	2	from_date	A	2838426	NULL	NULL	NULL	BTREE
3	salaries	1	idx_emp_no	1	emp_no	A	299318	NULL	NULL	NULL	BTREE

Debian [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Actividades MySQL Workbench 13 de ene 15:06

MySQL Workbench Local instance 3306

File Edit View Query Database Server Tools Scripting Help

Schemas Schemas

Tables

departments

Columns

dept_no dept_name

Indexes

Foreign Keys

Triggers

dept_emp

Object Info Session

Connection Details

Name: Localinstance 306

Host: localhost

Port: 3306

Login User: root

Current User: root@localhost

SSL cipher: SSL not used

Result Grid Filter Rows: Export: Wrap Cell Content:

#	id	select_type	table	partition	type	possible_keys	key	key_len	ref	rows	fil
1	1	SIMPLE	de	NULL	index	PRIMARY,dept_no	dept_no	16	NULL	331143	10
2	1	SIMPLE	d	NULL	eq_ref	PRIMARY,dept_name	PRIMARY	16	employees.de.dept_no	1	10
3	1	SIMPLE	s	NULL	ref	PRIMARY,`idx_emp_no`	PRIMARY	4	employees.de.emp_no	9	10

Result 4

ivana@debian: ~

```

# Time: 2025-01-13T14:02:03.348889Z
# User@Host: root[root] @ localhost [:1] Id:      9
# Query_time: 11.524787  Lock_time: 0.000114 Rows_sent: 9  Rows_examined: 380530
1
SET timestamp=1736776911;
SELECT d.dept_name, AVG(s.salary) AS avg_salary
FROM departments AS d
JOIN dept_emp AS de ON d.dept_no = de.dept_no
JOIN salaries AS s ON de.emp_no = s.emp_no
GROUP BY d.dept_name
LIMIT 0, 1000;
# Time: 2025-01-13T14:05:19.293790Z
# User@Host: root[root] @ localhost [:1] Id:      8
# Query_time: 0.000200  Lock_time: 0.000001 Rows_sent: 0  Rows_examined: 0
SET timestamp=1736777119;
SELECT st.* FROM performance_schema.events_stages_history_long st WHERE st.nesting_event_id = 148;
# Time: 2025-01-13T14:05:19.295047Z
# User@Host: root[root] @ localhost [:1] Id:      8
# Query_time: 0.000178  Lock_time: 0.000000 Rows_sent: 0  Rows_examined: 0
SET timestamp=1736777119;
SELECT st.* FROM performance_schema.events_waits_history_long st WHERE st.nesting_event_id = 148;
root@debian:/home/ivana#
OST: localhost          2 1 SIMPLE   d    NULL      eq_ref PRIMARY,dept_name PRIMARY 16      employees.
ort: 3306               3 1 SIMPLE   s    NULL      ref     PRIMARY,`idx_emp_no` PRIMARY 4       employees.
igin User: root
... 
```

