Optimización y rendimiento en MySQL

19

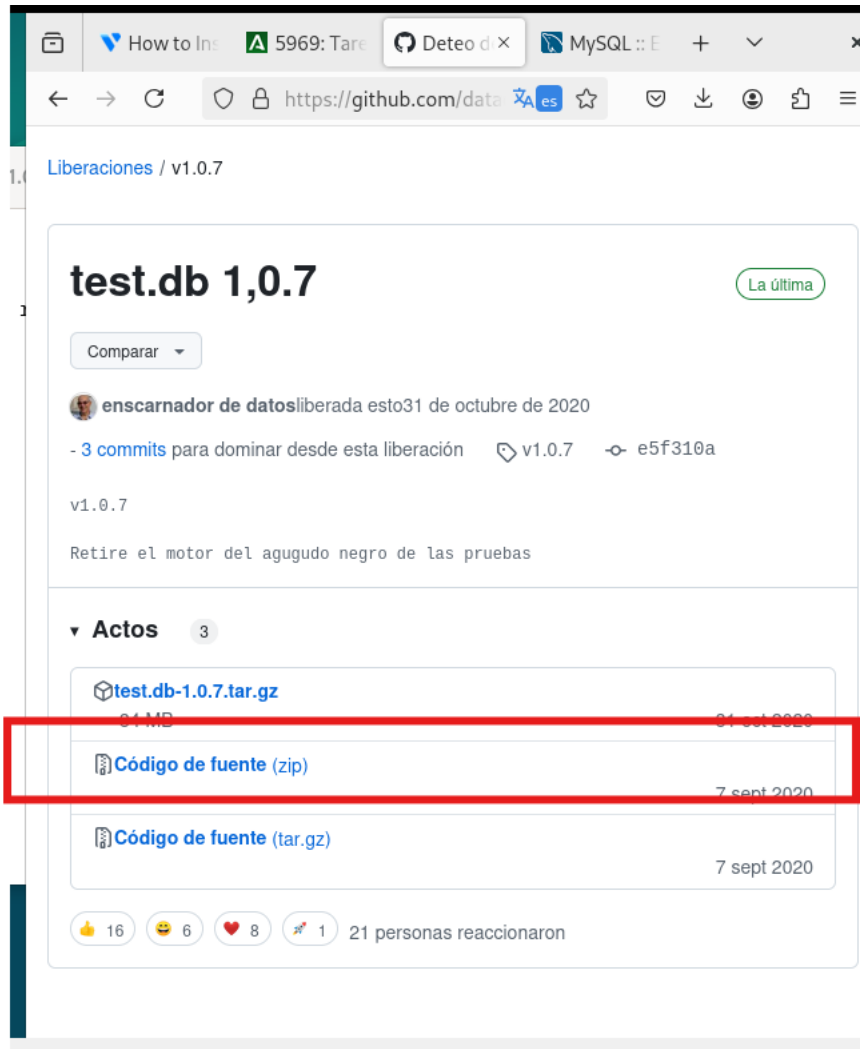
IVANA SÁNCHEZ PÉREZ-2ºASIR

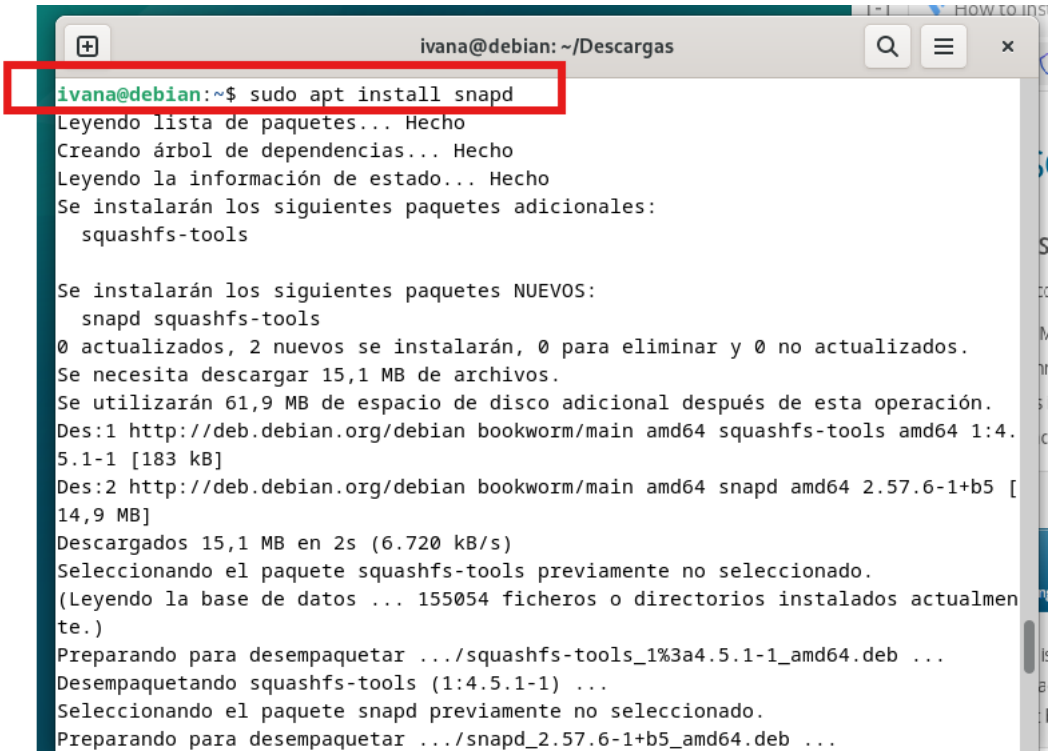
1.- Introducción

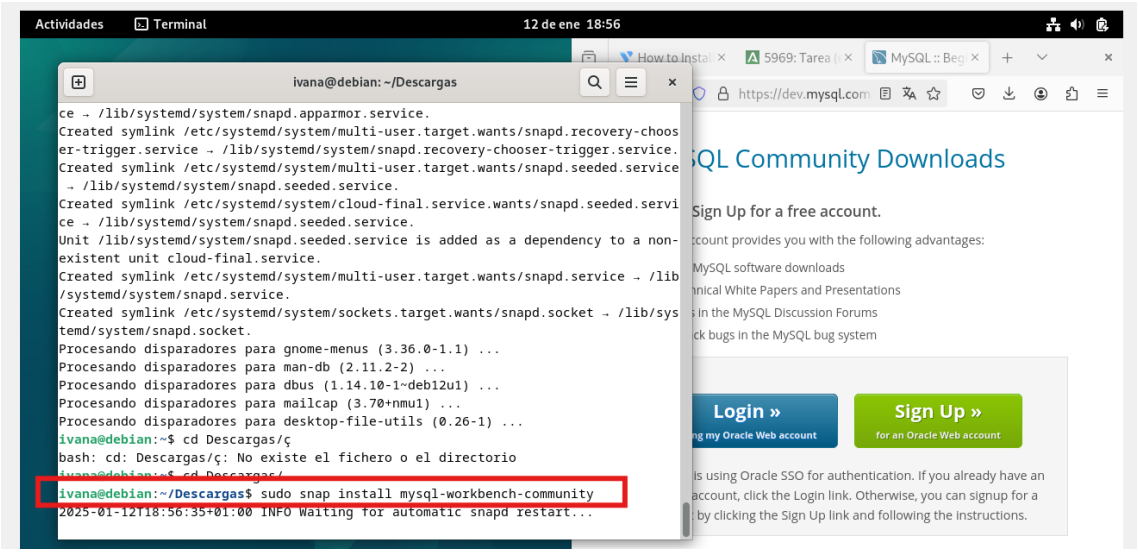
En esta tarea tiene como objetivo ejemplificar de forma práctica algunos de los aspectos más relevantes para que una base de dato se mantenga en marcha en las mejores condiciones de rendimiento.

2.- Apartado 1

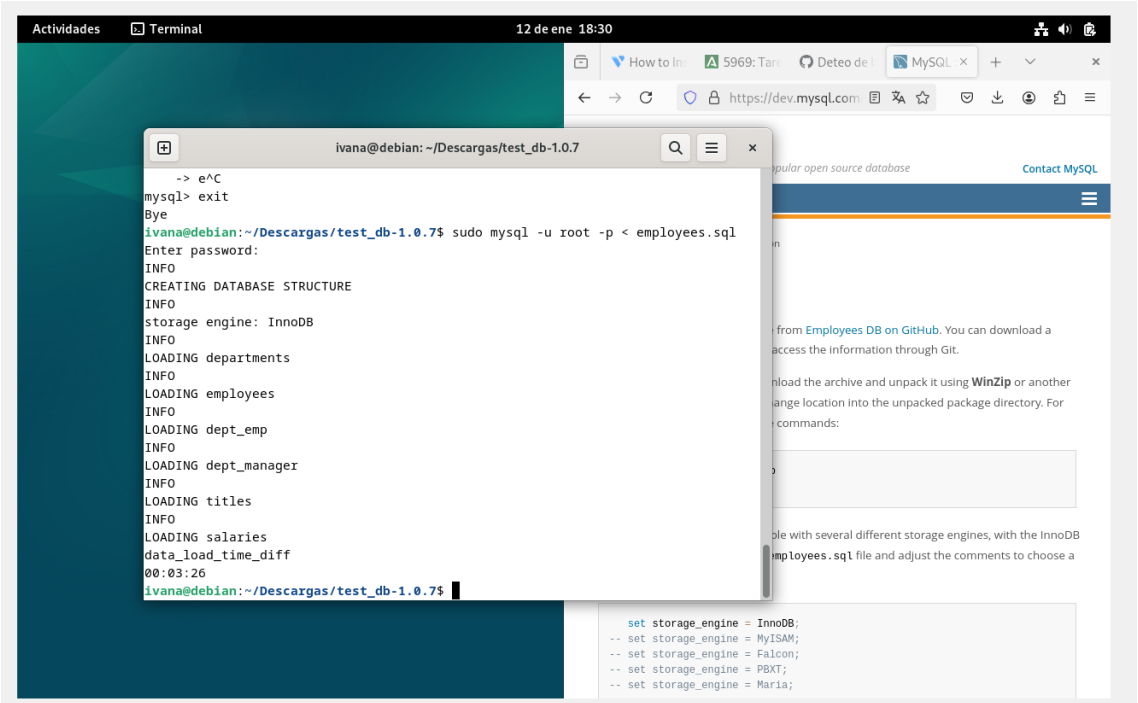
Voy a utilizar una máquina virtual Debian12 para realizar la tarea. Para ello, lo primero que voy a hacer es descargar MySQL a través de su terminal, con sus correspondientes archivos debug, y el MySQL Workbench, para trabajar de una forma más cómoda con una interfaz gráfica.

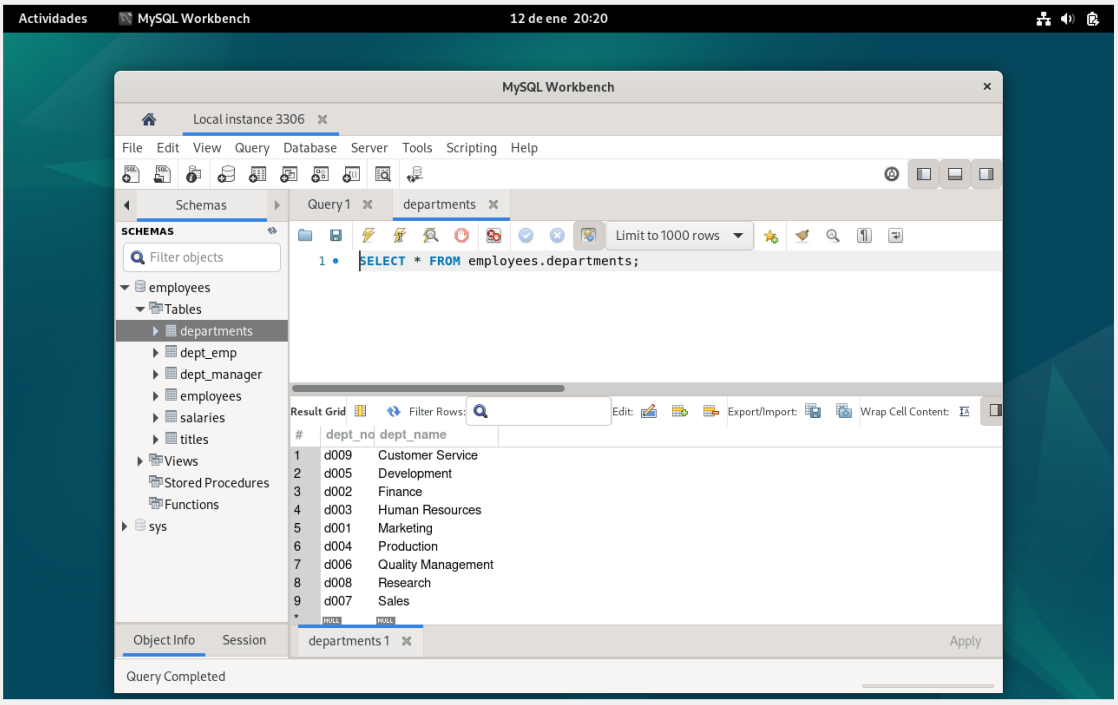






Una vez instalados, procedemos a descargarnos la base de datos de prueba que MySQL ha puesto a disposición en su web, llamada employees.sql, y que descargaremos en el terminal y comprobaremos que se ha creado a la misma vez en Workbench.

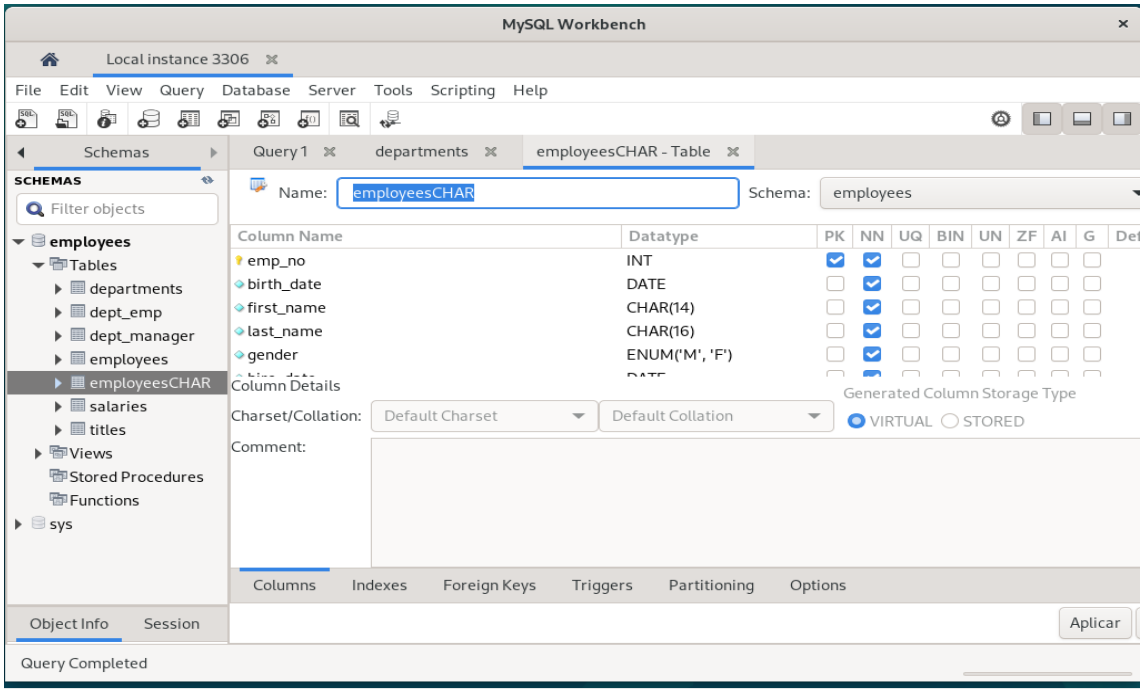

Actividades
Terminal
12 de ene 18:30
ivana@debian: ~/Descargas/test_db-1.0.7
-> e^C
mysql> exit
Bye
ivana@debian:~/Descargas/test_db-1.0.7$ sudo mysql -u root -p < employees.sql
Enter password:
INFO
CREATING DATABASE STRUCTURE
INFO
storage engine: InnoDB
INFO
LOADING departments
INFO
LOADING employees
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$
Contact MySQL
set storage_engine = InnoDB;
-- set storage_engine = MyISAM;
-- set storage_engine = Falcon;
-- set storage_engine = PBXT;
-- set storage_engine = Maria;

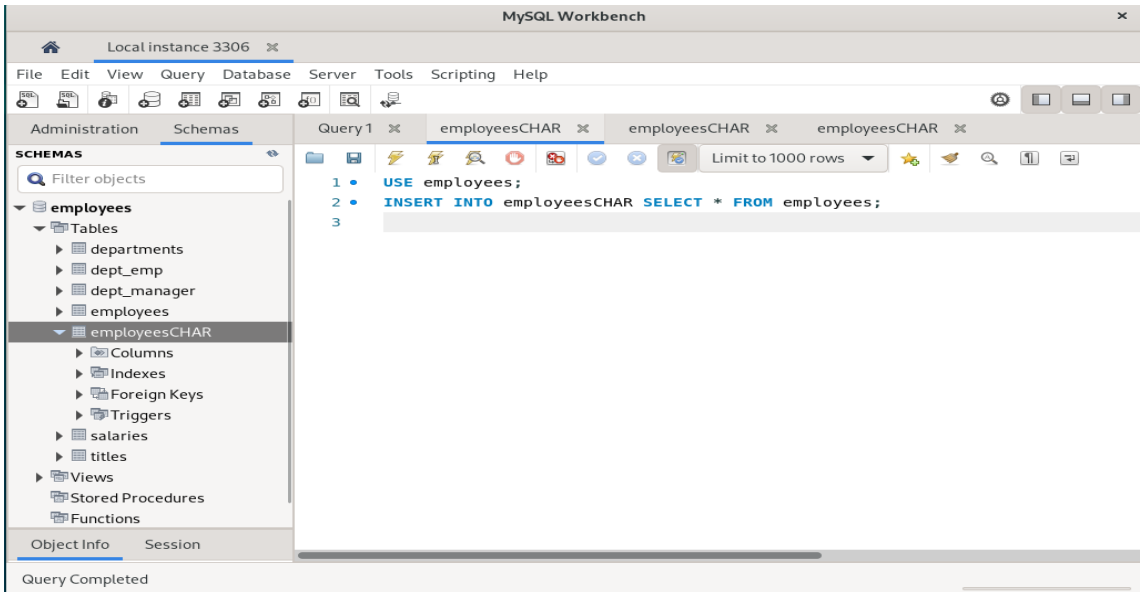

Actividades
MySQL Workbench
12 de ene 20:20
MySQL Workbench
Local instance 3306
File Edit View Query Database Server Tools Scripting Help
Schemas
Query 1
departments
SCHEMAS
Filter objects
employees
Tables
departments
dept_emp
dept_manager
employees
salaries
titles
Views
Stored Procedures
Functions
sys
Limit to 1000 rows
1 SELECT * FROM employees.departments;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
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

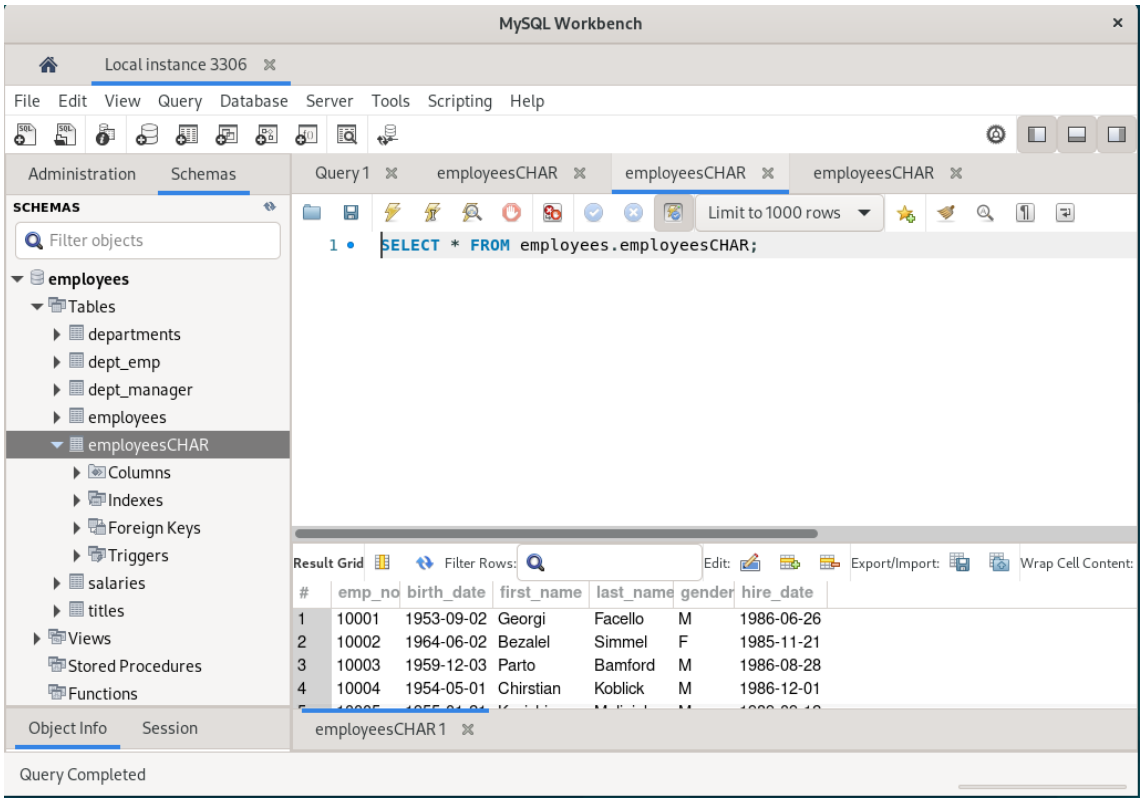
3.- Apartado 2

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

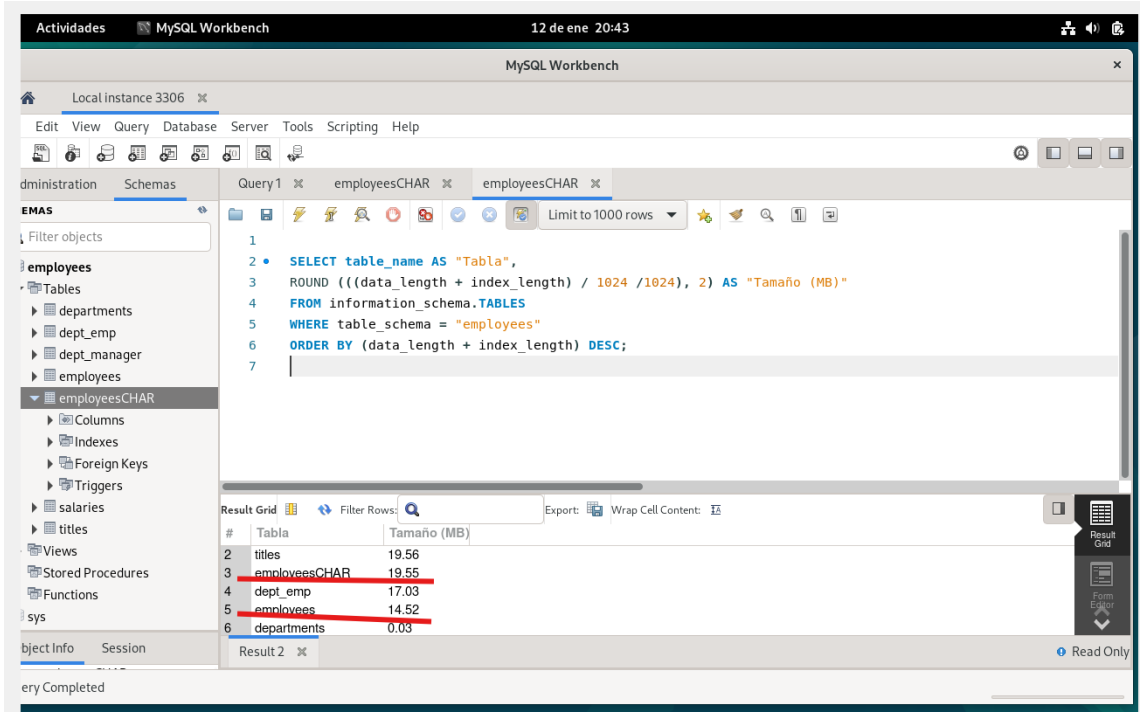
CHAR y VACHAR son tipos de datos que pueden almacenar cadenas de caracteres de diferentes longitudes. La diferencia entre ambas estriba en la forma en que asignan y manejan el espacio para las cadenas. CHAR significa cadena de caracteres de longitud fija, mientras que VARCHAR significa cadena de caracteres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







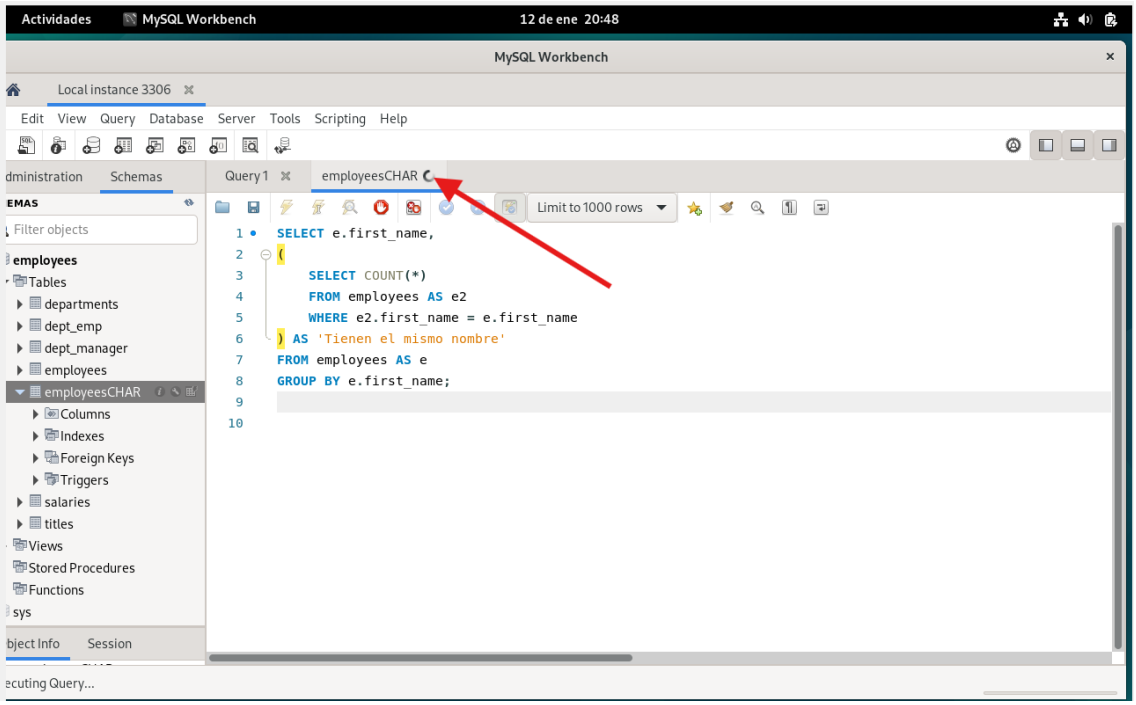
Se observará que hay una diferencia de aproximadamente 5MB entre las dos tablas.



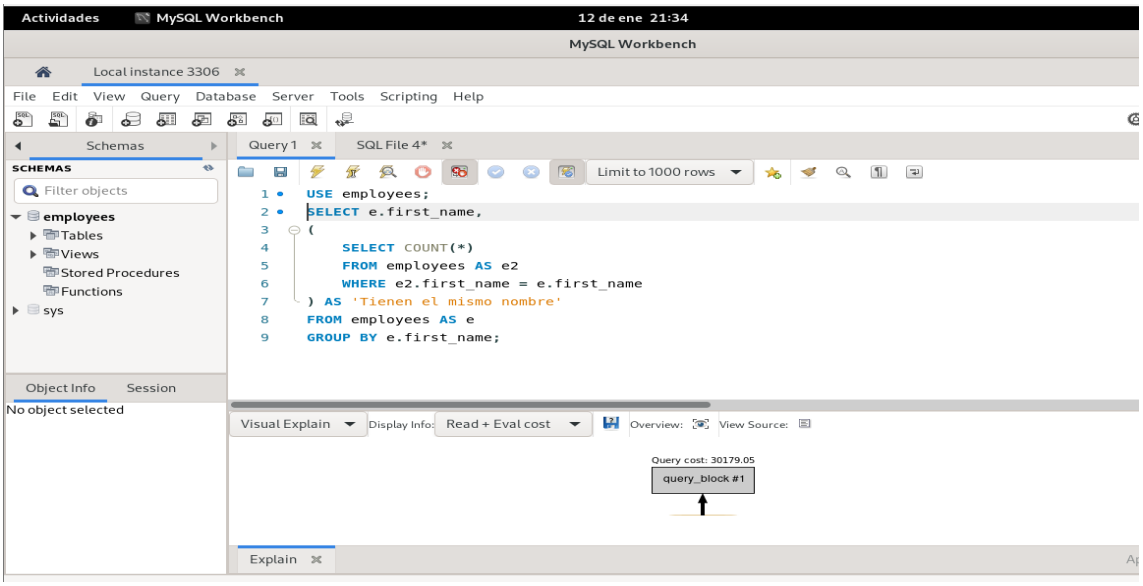
4.- APARTADO 3

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

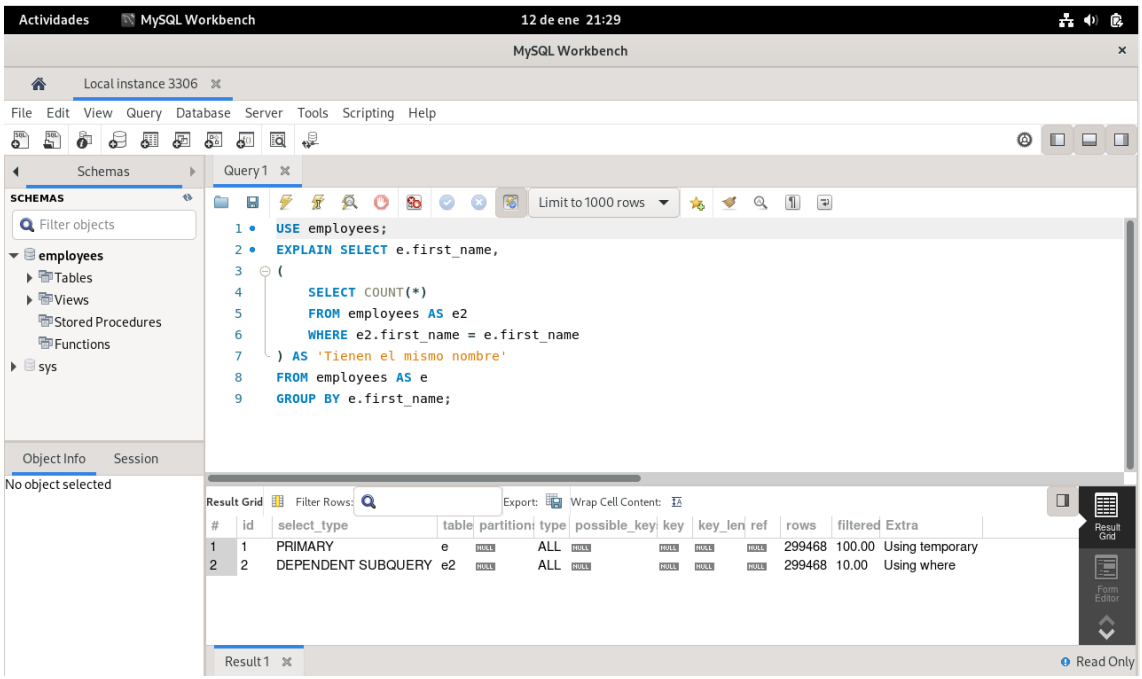
Primero hacemos una consulta normal pidiendo un listado con los nombres de los empleados y la cantidad de ellos que tienen el mismo nombre (Consulta pesada).



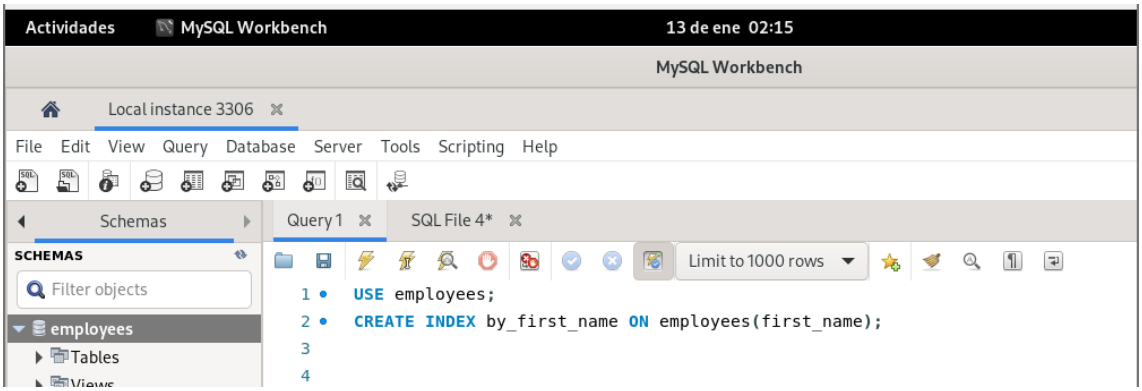
El tiempo de ejecución se dispara a los 1000 segundos que tenemos por defecto como límite. El resultado final es un error.

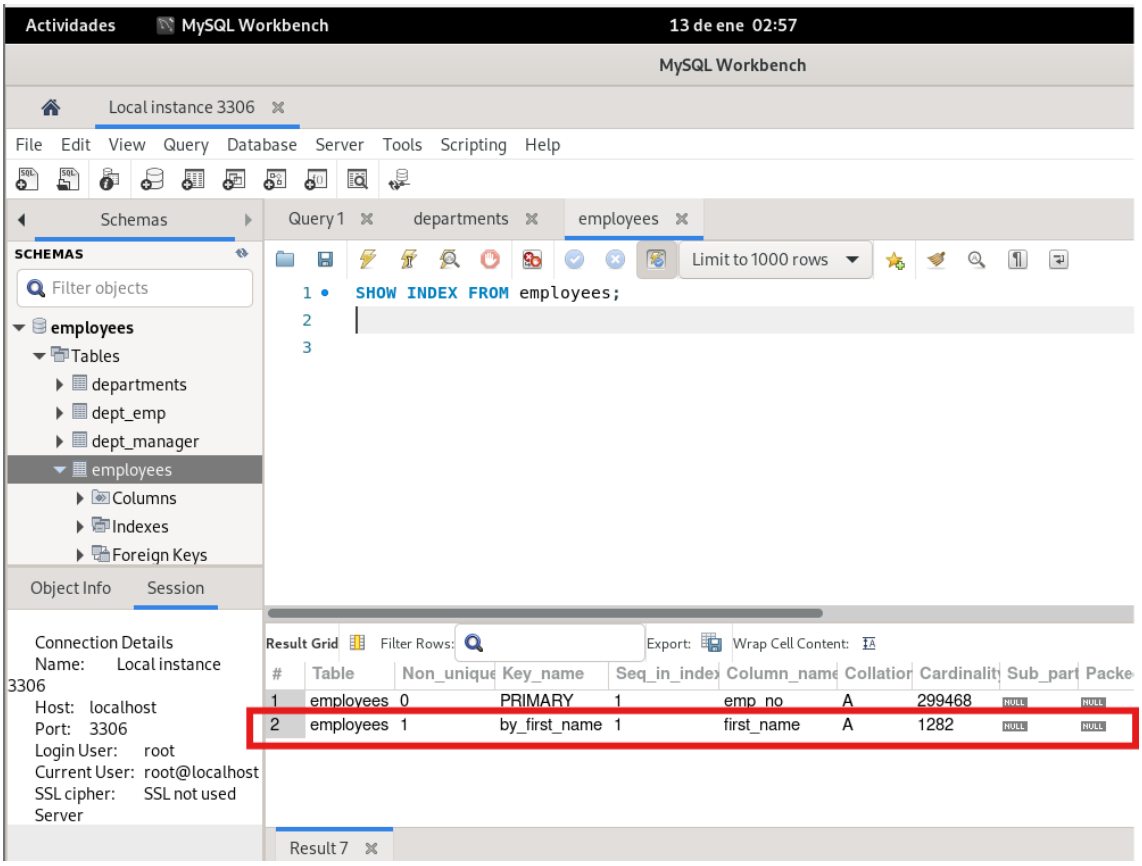


Aplicaremos la sentencia EXPLAIN delante de SELECT y observamos que no sólo no da error, sino que da un resultado inmediato.

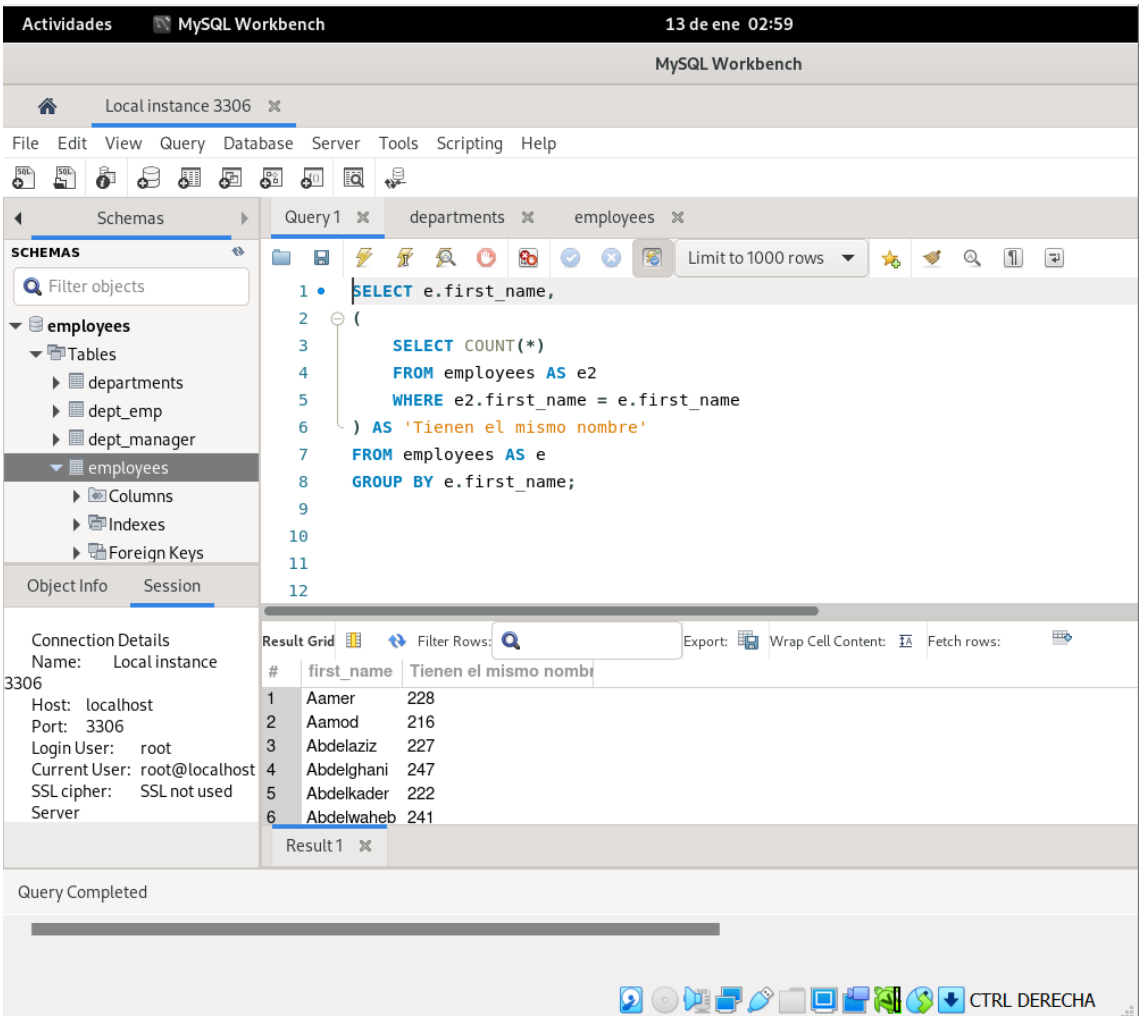


Ahora vamos a crear un índice y ejecutamos una consulta. Esto hará que ésta se ejecute en menos de 1 segundo.

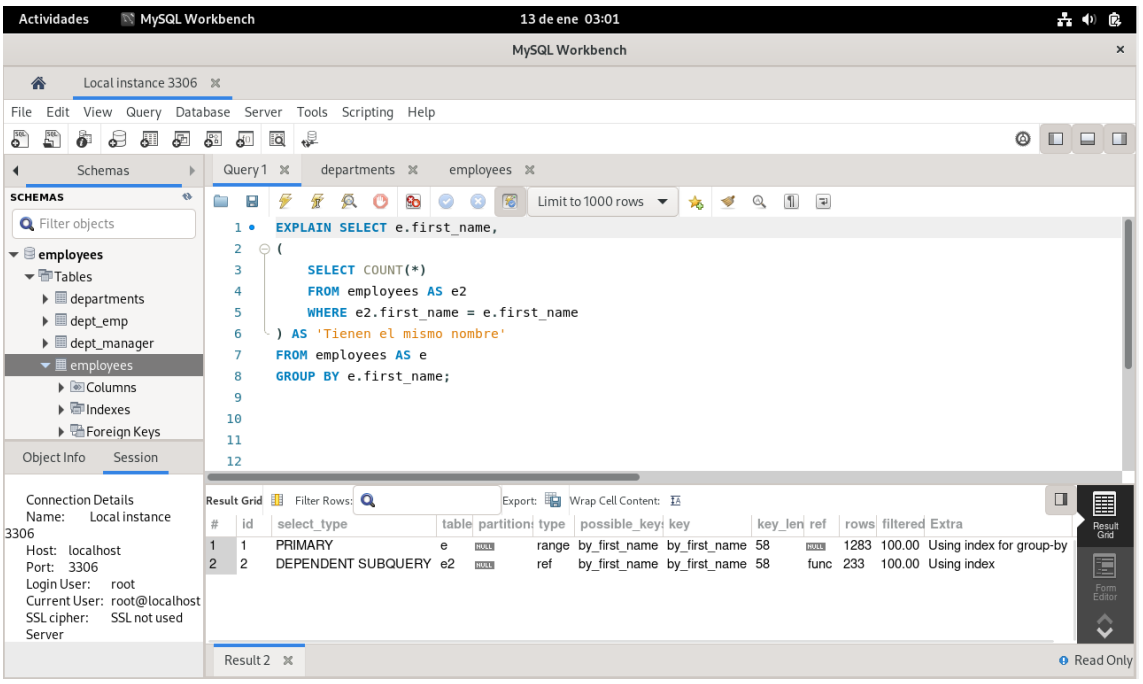




Hacemos una consulta con el índice.



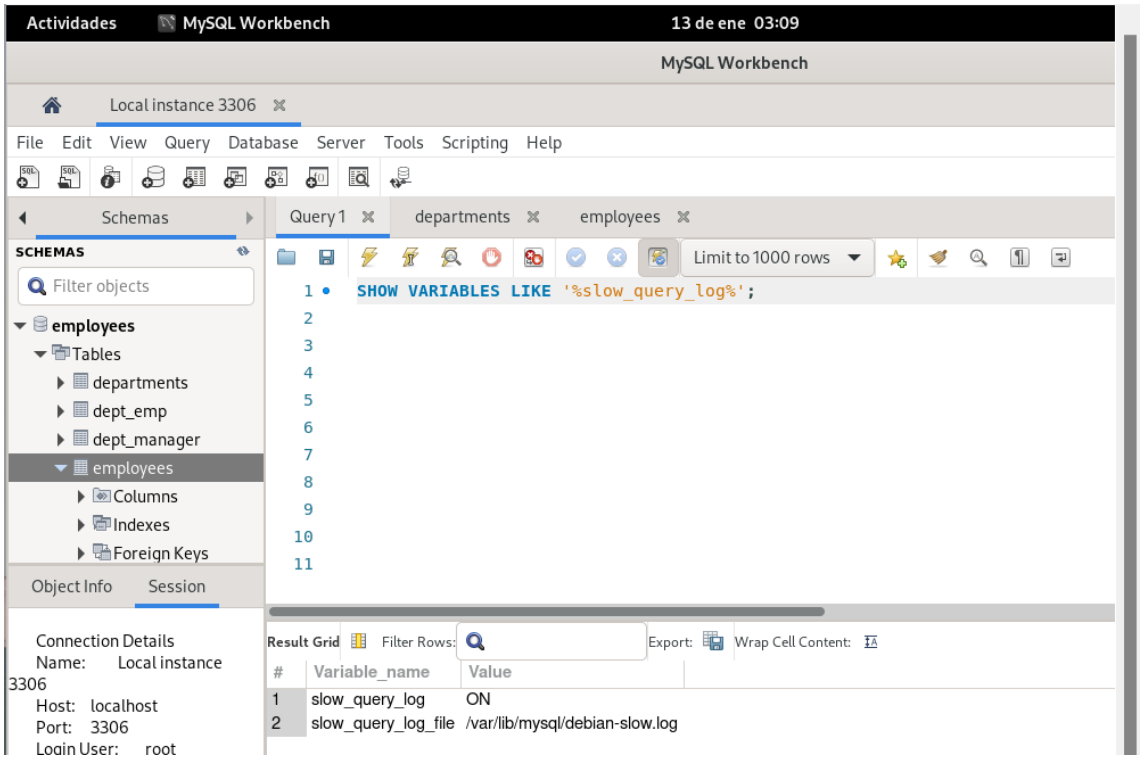
Y una segunda consulta con EXPLAIN

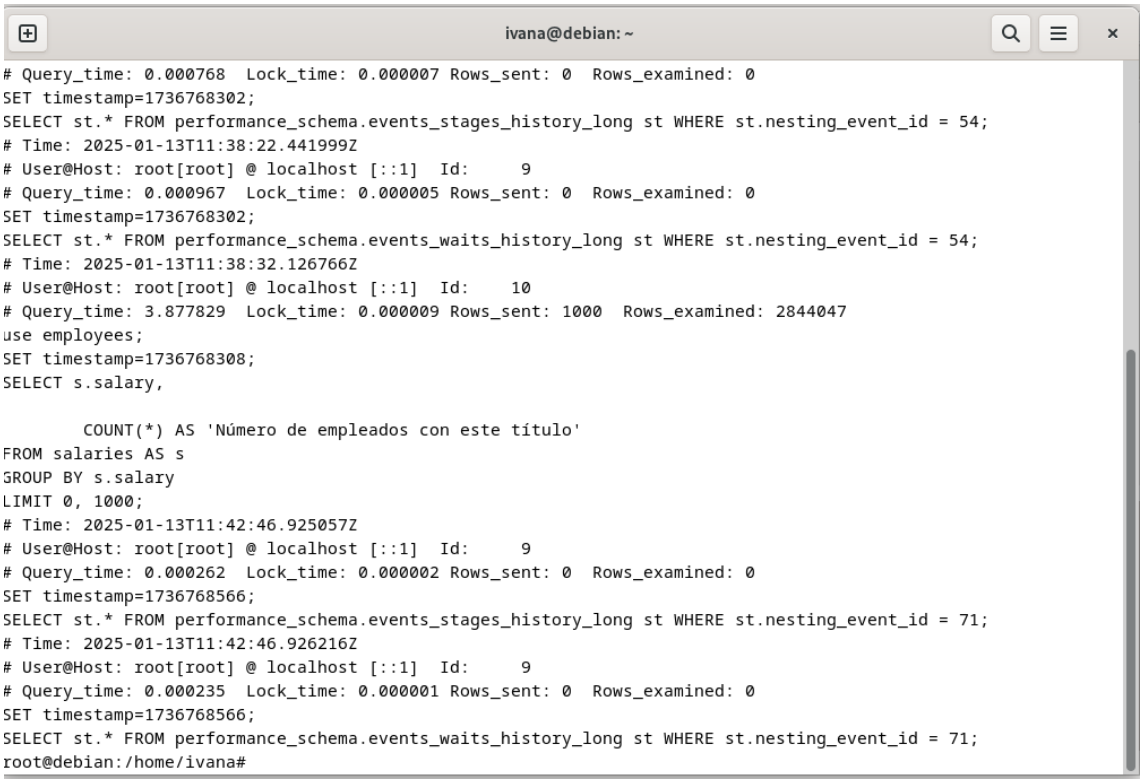


5.- Apartado 4

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.

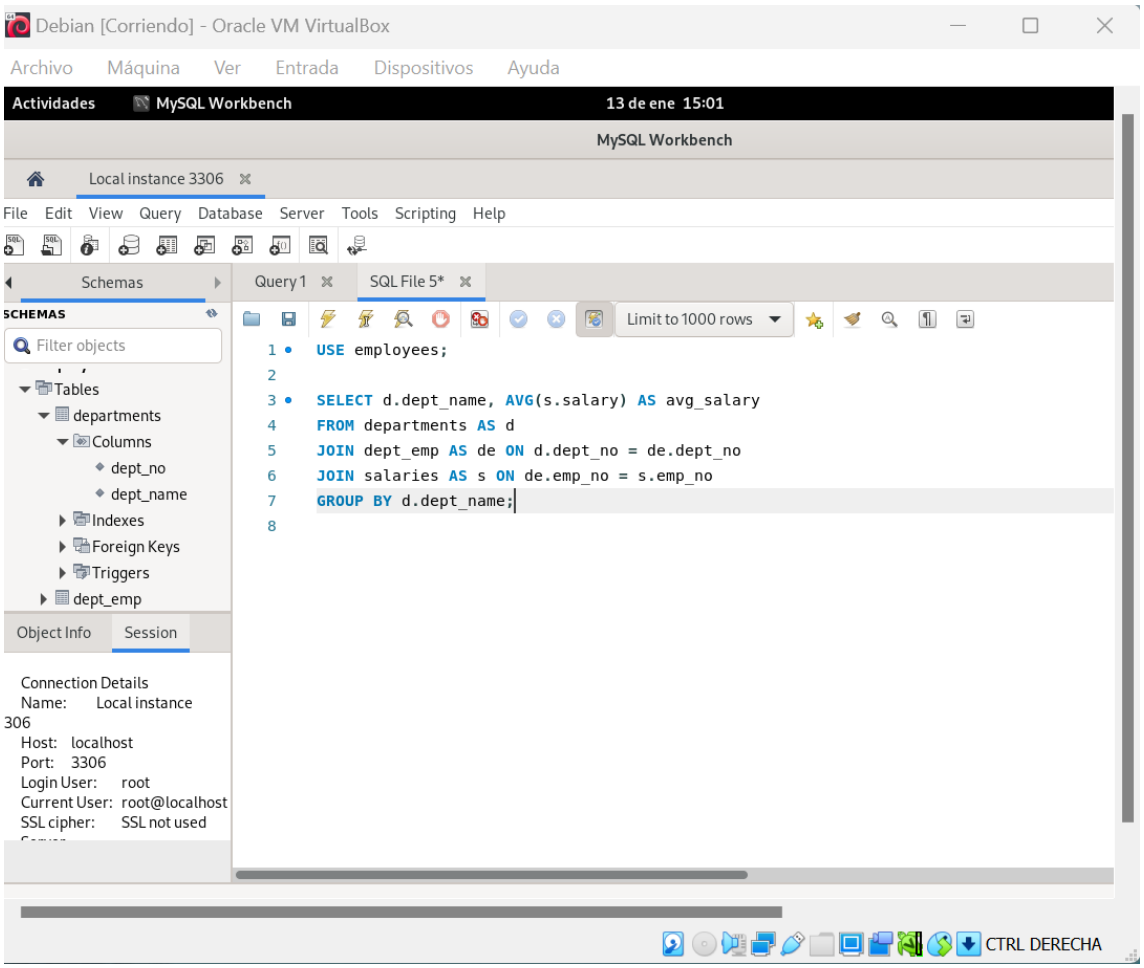




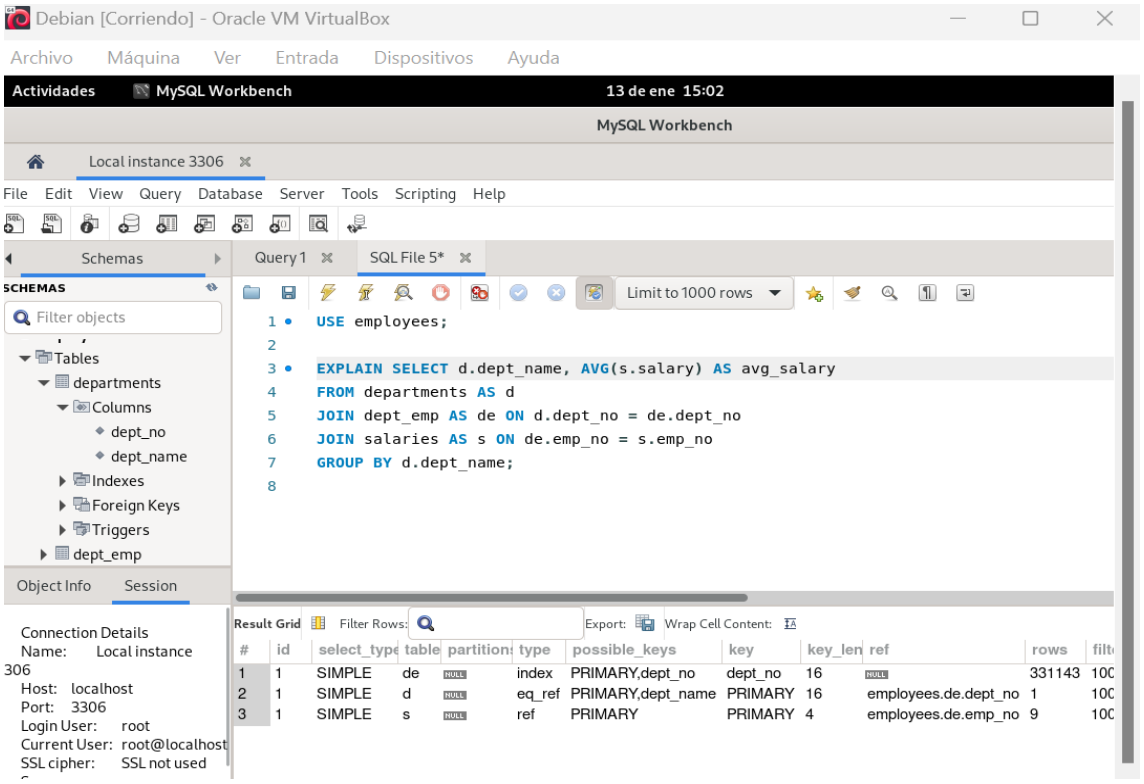
6.- Apartado 5

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

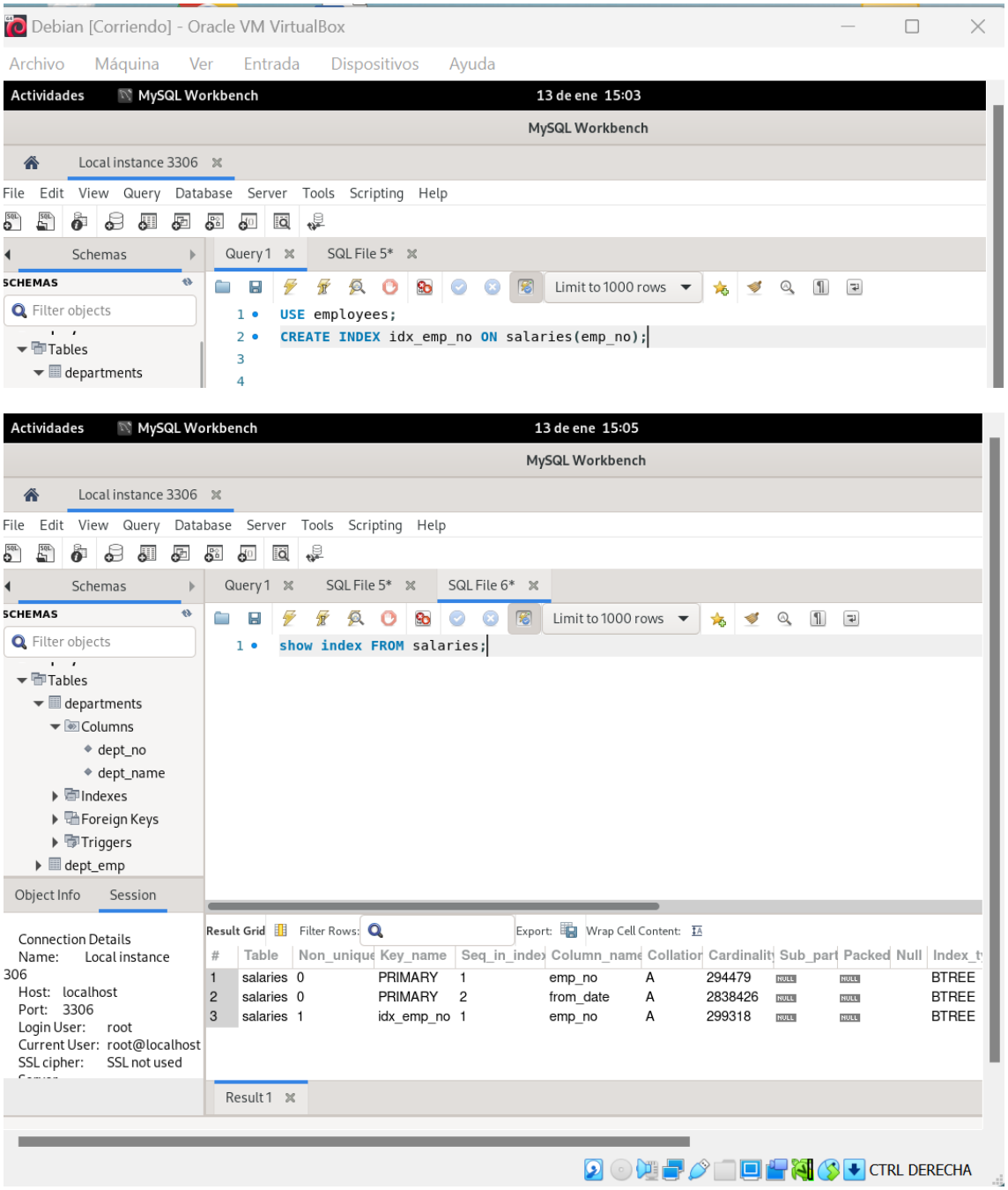
a) Consulta inicial

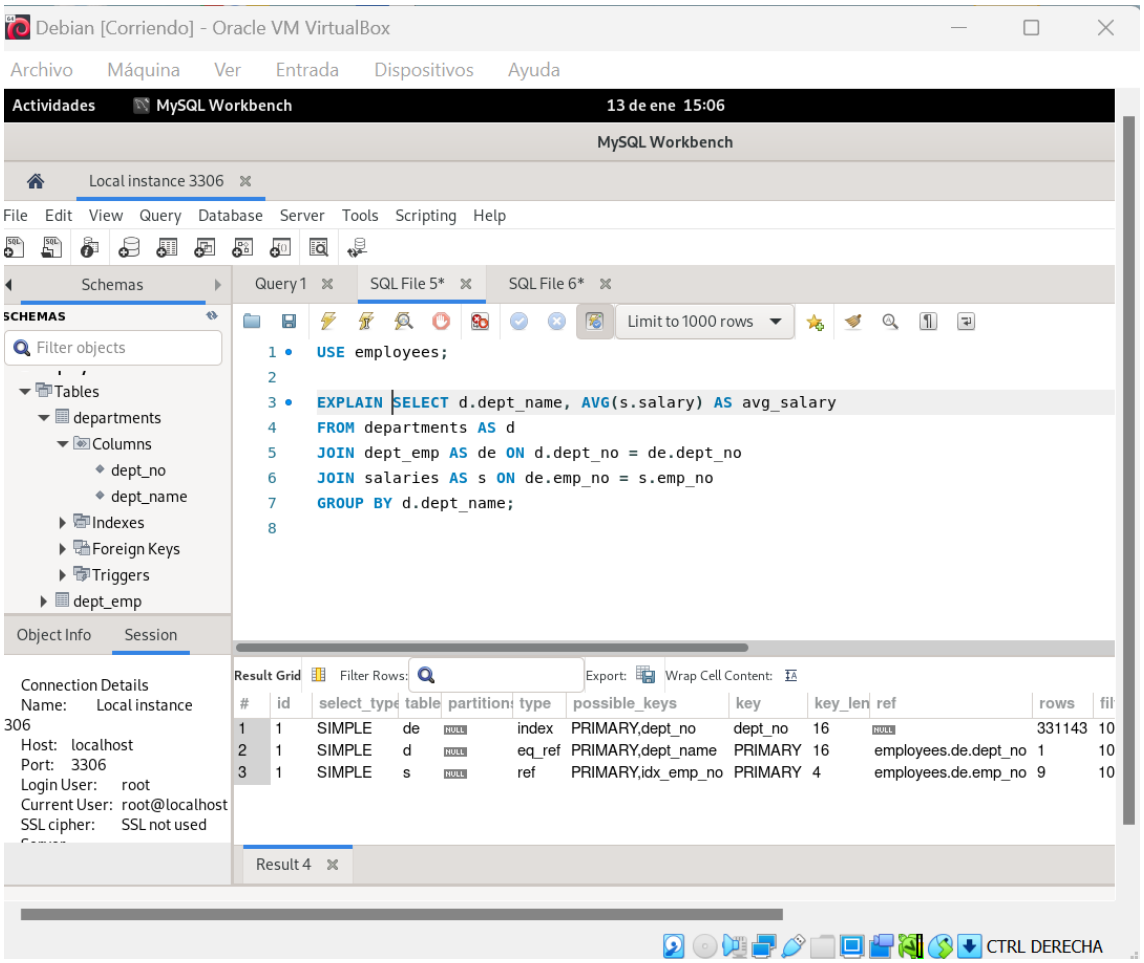


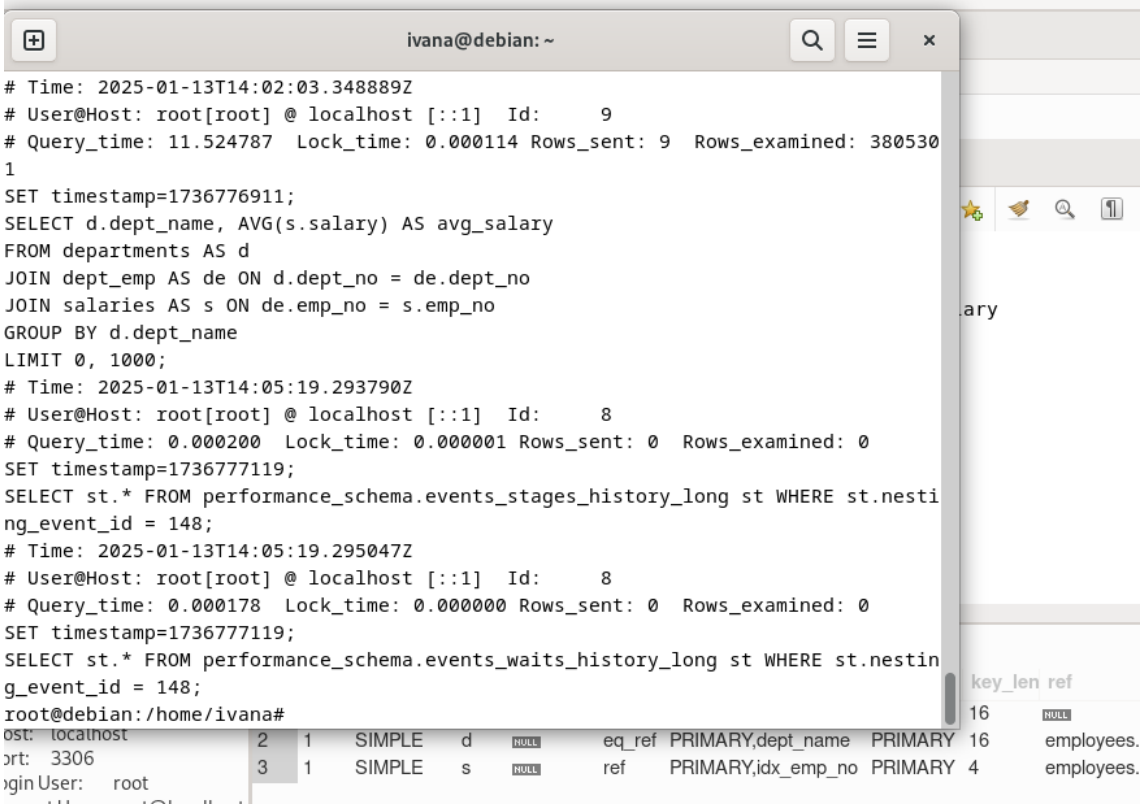
b) Consulta con el comando EXPLAIN



c) Creación de un índice




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
Query 1 SQL File 5* SQL File 6*
Limit to 1000 rows
1 • USE employees;
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;
Tables
departments
Columns
dept_no
dept_name
Indexes
Foreign Keys
Triggers
dept_emp
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
Result Grid Filter Rows: Export: Wrap Cell Content:
id select_type table partitions 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
CTRL DERECHA


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.nesti
ng_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.nestin
g_event_id = 148;
root@debian:/home/ivana#
key_len ref
16 NULL
2 1 SIMPLE d NULL eq_ref PRIMARY,dept_name PRIMARY 16 employees.
3 1 SIMPLE s NULL ref PRIMARY,idx_emp_no PRIMARY 4 employees.