Práctica 4

Primera parte

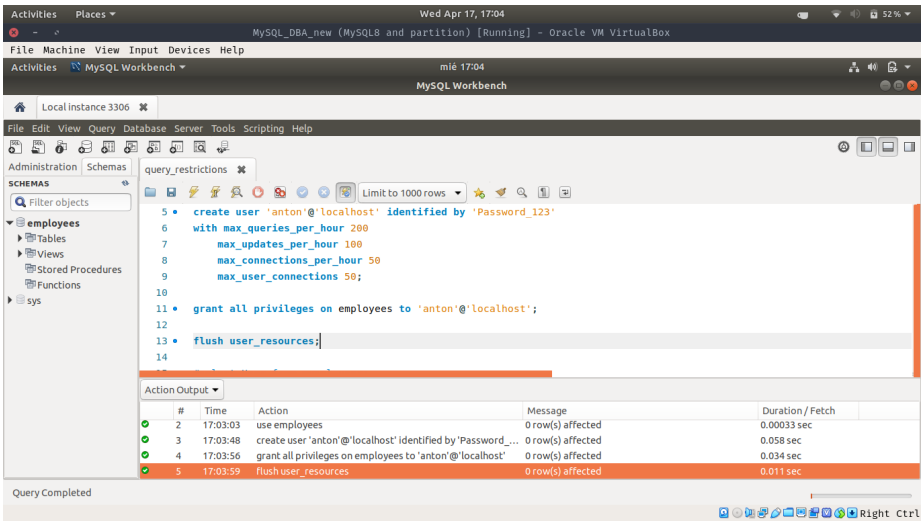

Figure 1: Creación y permisos dados a usuario anton

Lo primero es crear el usuario anton con contraseña “Password_123”.

A este usuario se le otorgaron 4 permisos:

1. Cantidad máxima de queries por hora (200).
2. Cantidad máxima de “updates” por hora (100).
3. Cantidad máxima de conexiones por hora (50).
4. Cantidad máxima de conexiones simultáneas (50).

El primer permiso (queries por hora) define la cantidad máxima de queries por hora, entendiéndose como queries cualquier consulta a la base de datos, es decir, “select”, “update”, “delete”, “create”, etc.

El segundo permiso (“updates” por hora) indica precisamente eso; número de queries “update” que tiene permitidos en una hora.

El tercer permiso (conexiones por hora) se refiere a la cantidad de veces que dicho usuario se puede conectar y desconectar a la DB.

Por último, en el cuarto permiso (cantidad máxima de conexiones) se indica la cantidad de conexiones simultáneas con la DB. Ejemplo de esto podría ser: estar conectado a través de varias terminales, en algún programa como phpMyAdmin o MySQL Workbench, etc., al mismo tiempo.

Cabe aclarar que cada que se los permisos negaran alguna acción al usuario “anton”, se tuvo que realizar un flush de los privilegios de los usuarios, como se muestra en la línea 13 de la figura 1. De esta manera no se tendría que esperar a que los permisos se restablecieran después de una hora y así poder continuar con la práctica.

Cantidad máxima de queries por hora

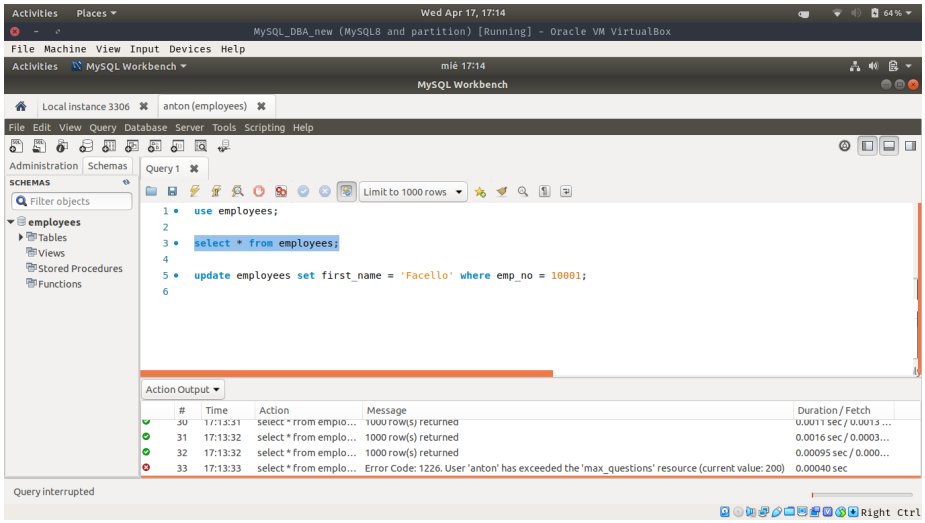

Figure 2: Cantidad máxima de queries por hora

Como se muestra en la figura 2, la base de datos solo permitió un total de 32 queries de los cuales una es un “use” y los restantes un simple “select”.

Cantidad máxima de “updates” por hora

Como se puede obsevar en la figura 3, la base de datos sí permitió los 100 updates, sin contar que la queries 1 fue un “use”.

Cantidad máxima de conexiones por hora

En esta parte se hizo un conexión-desconexión un total de 6 veces antes de que se mostrara el error que aparece en la figura 4. Este erro indica que la cantidad máxima de queries se ha agotado.

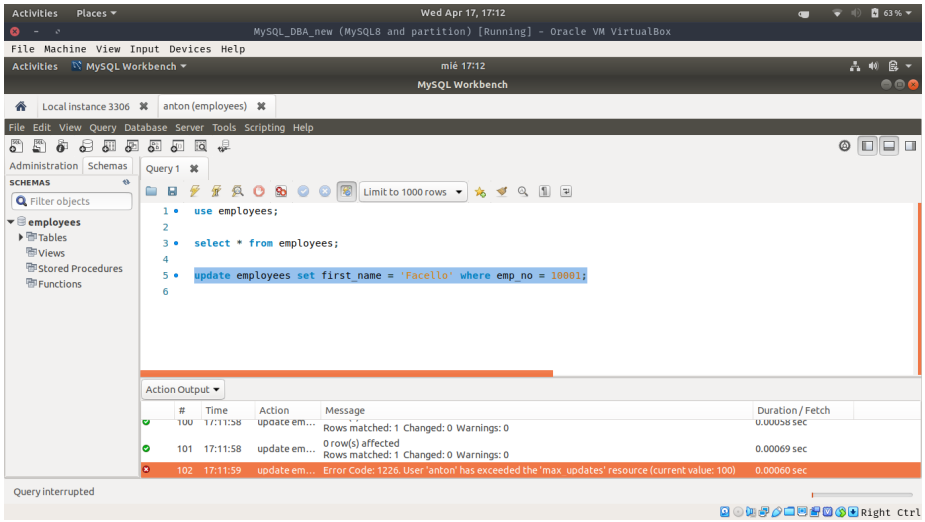

Figure 3: Cantidad máxima de “updates” por hora

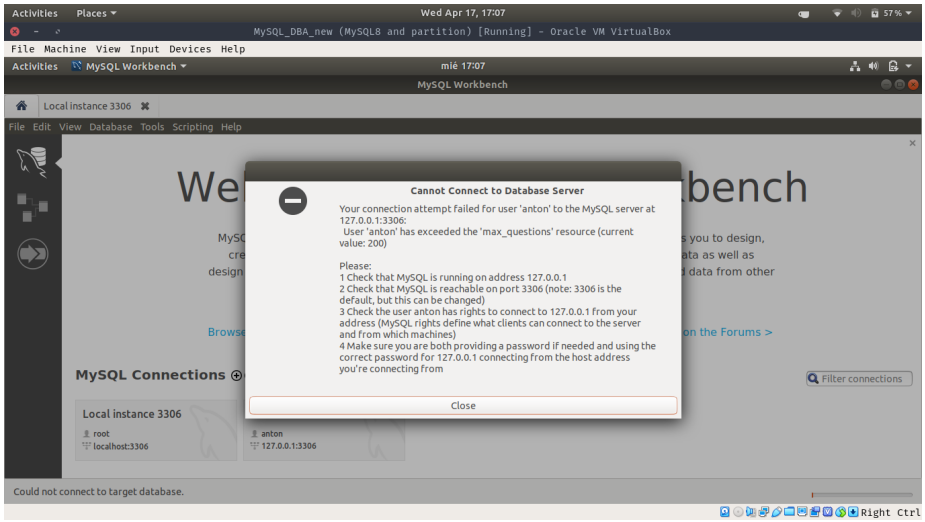

Figure 4: Cantidad máxima de conexiones por hora

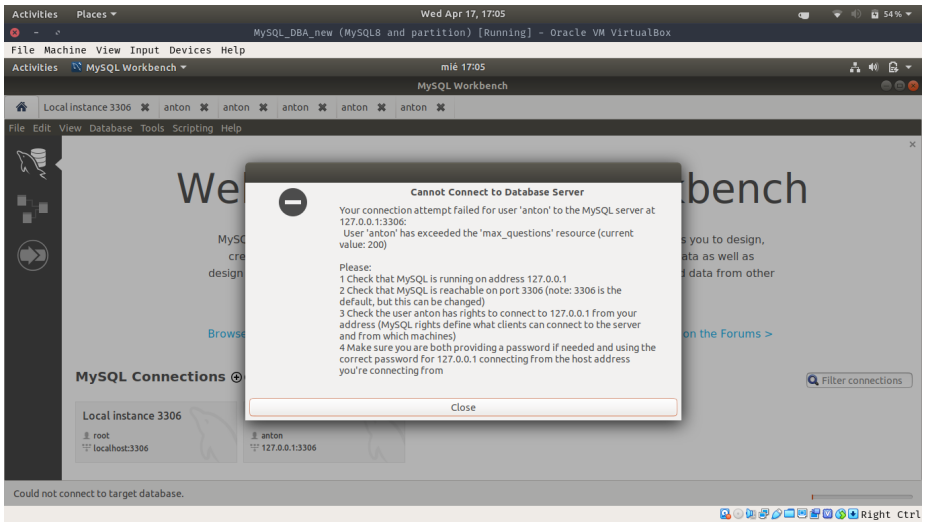

Figure 5: Cantidad máxima de conexiones simultáneas

Cantidad máxima de conexiones simultáneas

La conexión de la base de datos solo permitió 5 conexiones simultáneas antes de arrojar un error al sexto intento. Las conexiones simultánes se pueden observar a manera de tabs en la parte superior de la figura 5.

Conclusión de la primera parte

Prácticamente ninguno de los permisos se cumplieron como se especificaron al momento de crear el usuario. Esto se debe a que MySQL Workbench requiere más conexiones adicionales para que pueda funcionar de manera correcta, sin embargo, hace uso de conexiones excesivas por lo que no se ve reflejada la configuración de los permisos, a excepción de los “updates” por hora.

Segunda parte

Creación de la base de datos

Lo primero que se realizó fue la creación de una base de datos llamada “app”, con el fin de que pudiese ser utilizada para esta práctica.

Creación de usuarios

Como se muestra en la figura 6, fueron creados 3 usuarios: un usuario llamado "desarrollador" en localhost, un usuario llamado "administrador" en localhost y otro usuario también llamado "administrador" pero que puede acceder desde cualquier host. La contraseña debió ser cambiada ya que las restricciones de contraseña por defecto de MySQL son más esctrictas, de otro modo arrojaría un error.

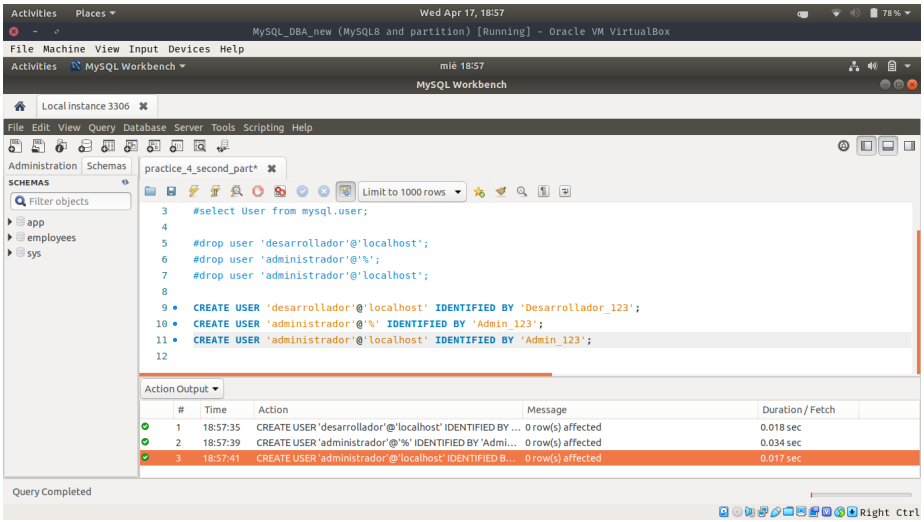

Figure 6: Creación de usuarios para asignar privilegios

Asignación de privilegios

El siguiente paso fue añadirle privilegios a los usuarios creados en el paso anterior, por lo que se ejecutaron las queries mostradas en el la figura 7.

Al usuario "administrador" que puede acceder desde cualquier host se le asignaron todos los privilegios posibles en todas las bases de datos y tablas, al usuario "administrador" que puede acceder únicamente desde localhost se le asignaron los permisos RELOAD y PROCESS en todas las tablas de todas las bases de datos y por último, al usuario "desarrollador" se le asignaron permisos en todas las tablas de la base de datos "app".

Creación de roles

El siguiente paso fue la creación de 3 roles: "desarrollador", "lectura" y "escritura", como se muestra en la figura 8.

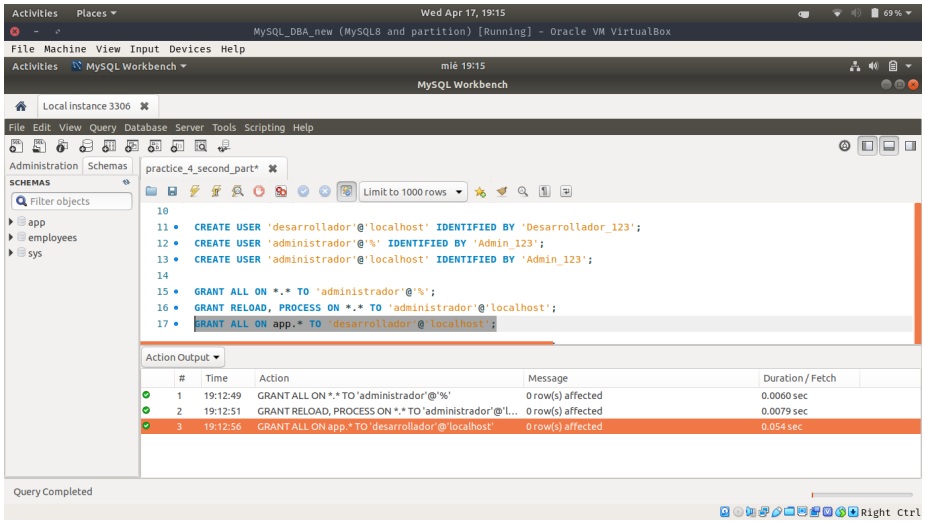

Figure 7: Asinación de privilegios a los usuarios creados

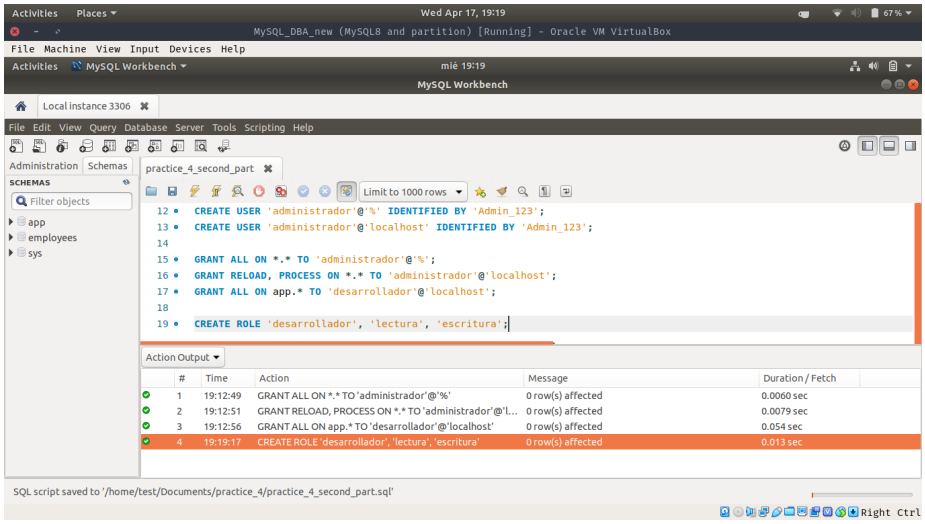

Figure 8: Creación de roles

Asignación de permisos a roles

Lo siguiente fue asignar los permisos a los roles creados en el paso anterior, como se muestra en la figura 9.

Al rol desarrollador se el asignaron permisos en todas las tablas de la base de datos “app”, al rol “lectura” se le asignaron igualmente permisos en todas las tablas de la base de datos “app” y por último al rol “escritura” se le asignaron permisos de “insert”, “update” y “delete” en todas las tablas de la base de datos “app”.

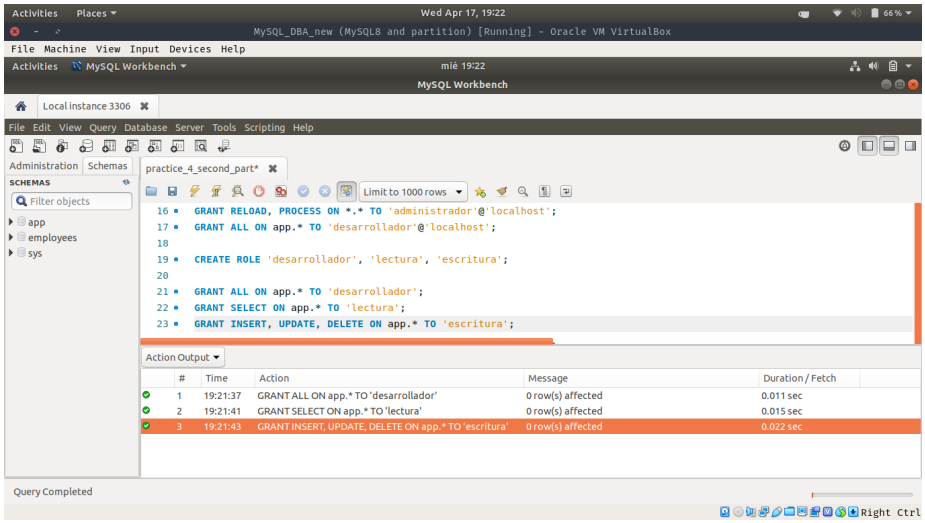

Figure 9: Asignación de los permisos a los roles creados

Creación de nuevos usuarios

Lo siguiente en la práctica fue crear 4 nuevos usuarios los cuales serán usados para asignarles roles.

Todos los usuarios puede acceder únicamente desde localhost. Se creó un usuario llamado “desarrollador1”, otro llamado “usuario_r1”, otro “usuario_r2” y un último llamado “usuario_wr”, como se muestra en la figura 10.

Asignación de roles a usuarios

El siguente paso fue asignar los roles, ya con privilegios asignados, a los nuevos usuarios creados anteriormente.

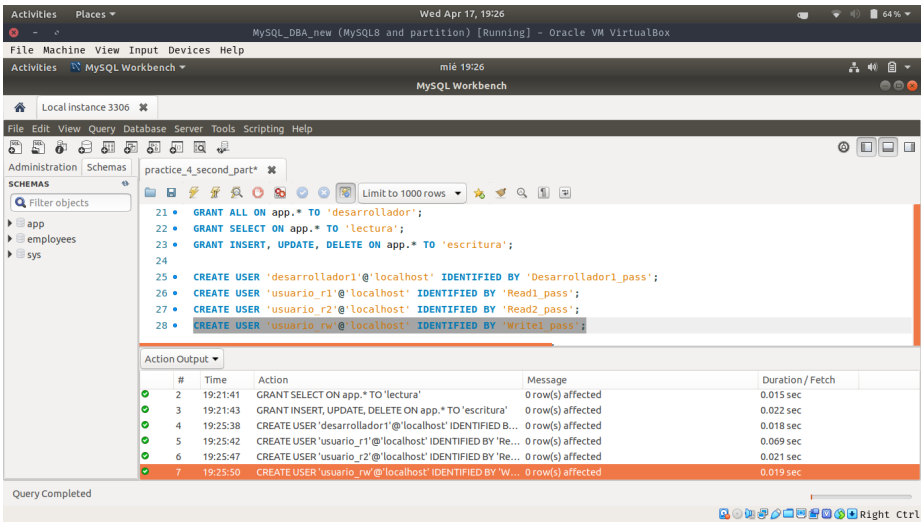

Figure 10: Creación de nuevos usuarios para asignarles roles

Recordar que todos los nuevos usuarios creados puede acceder solo desde localhost. Al usuario "desarollador1" se le asignó el rol "desarollador", a los usuarios "usuario_r1" y "usuario_r2" se les asignaron el rol "lectura" y por último, al usuario "usuario_rw" se le asignó el rol "escritura". Esto se puede apreciar en la figura 11.

Mostrar privilegios y roles de un usuario

También se pueden asignar privilegios a los usuarios (aparte de los roles), sin embargo, roles y privilegios no pueden ser asignados en la misma sentencia por lo que, si se desea asginar roles y privilegios a un usuario, se tiene que realizar por separado. La figura 12 muestra la query utilizada para mostrar tanto los privilegios como los roles asignados a un usuario.

Mostrar lista detallada de privilegios de un usuario con rol/roles específicos

En el paso anterior no muestra información detallada sobre los privilegios del usuario de interés, sin embargo, la query mostrada en la figura 13 nos muestra los privilegios que tiene un usuario con un rol específico.

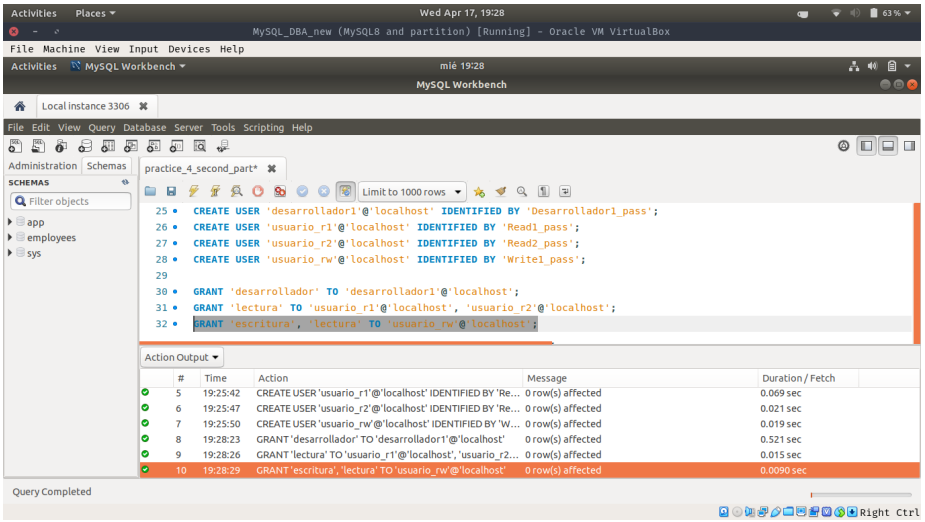

Figure 11: Asignación de roles a los nuevos usuarios creados

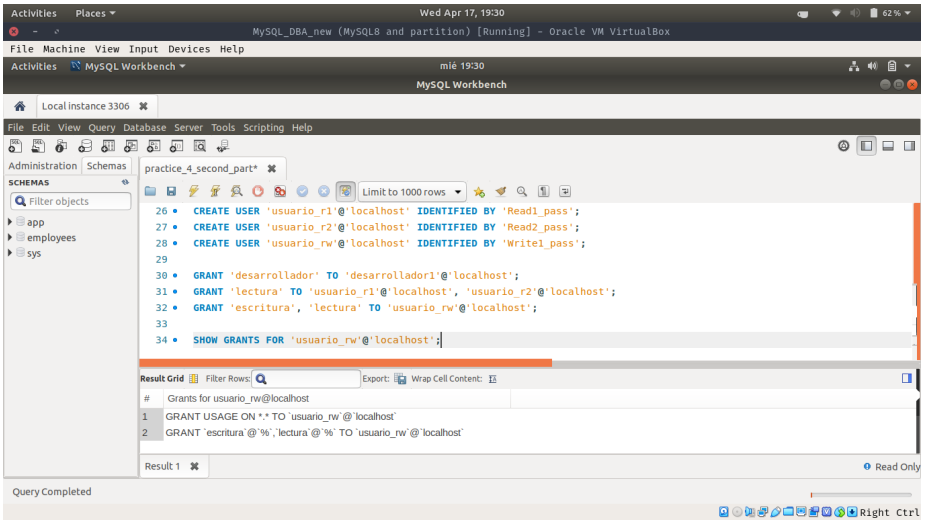

Figure 12: Mostrar privilegios y roles de un usuario

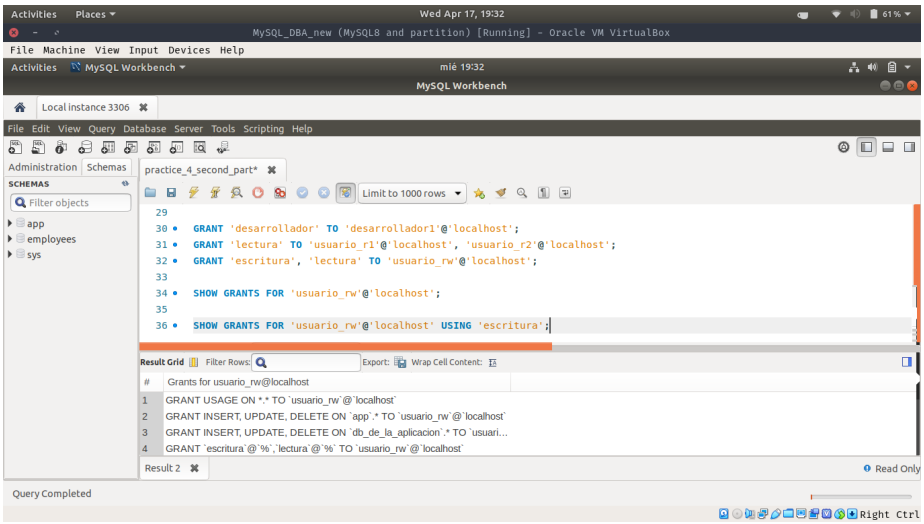

Figure 13: Mostrar privilegios y roles de un usuario con un rol específico

REVOKE y la variable @@mandatory_roles

A aquellos usuarios que se encuentren en la variable @@mandatory_roles no se les pueden revocar los privilegios. En la figura 14 se muestra que ninguno de los usuarios creados a lo largo de este práctica se encuentran ahí; a todos los usuarios que hemos creado se les pueden revocar los privilegios.

Revocar privilegios a un rol

En la figura 15 se muestra como se remueven los privilegios de “insert”, “update” y “delete” en todas las tablas de la base de datos “app” al rol “escritura”. Esto suele ser útil cuando se necesita remover temporalmente los permisos a un conjunto de usuarios ya que esta query también afectará a los usuarios que tengan asignado el rol a revocar.

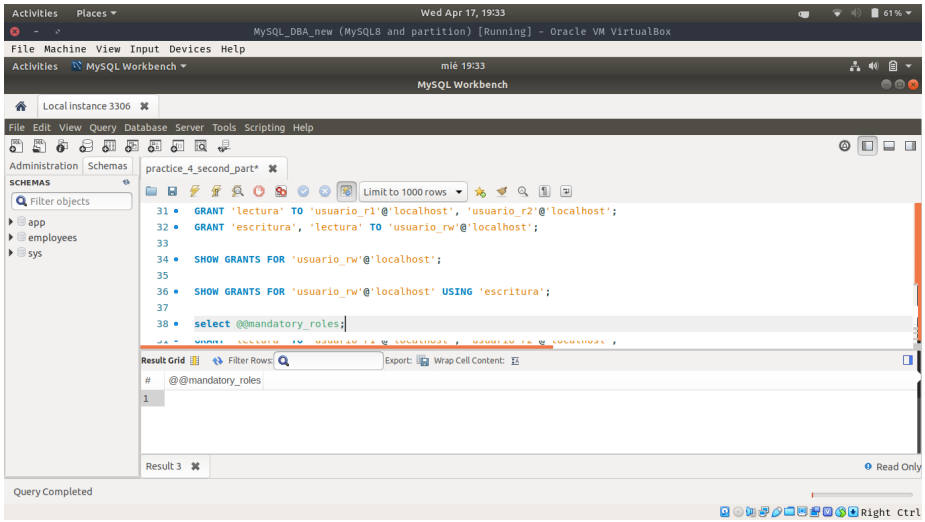

Figure 14: Contenido de la variables @@mandatory_roles

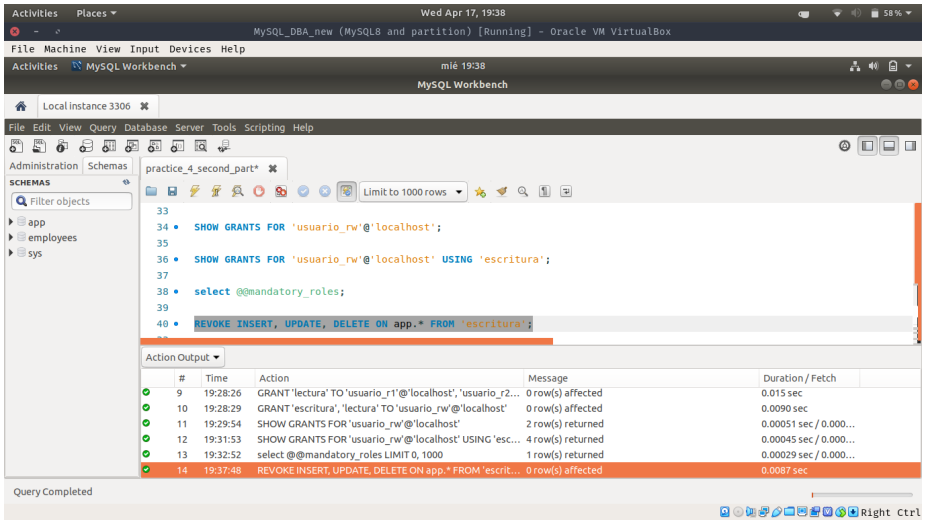

Figure 15: Revocar permisos a un rol