

UNIVERSIDAD SAN CARLOS DE GUATEMALA

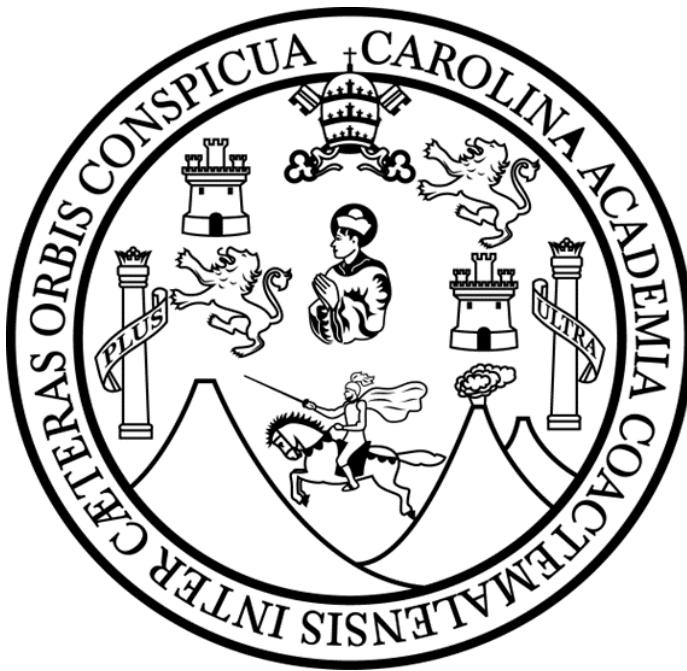
FACULTAD DE CIENCIAS DE LA INGENIERÍA

BASES DE DATOS 2

ING. MARLON FRANCISCO ORELLANA LOPEZ

AUXILIAR: JHONATHAN DANIEL TOCAY COTZOJAY

SECCIÓN P



PRÁCTICA 1

GRUPO #16

POR:

201930697 3348 21282 0901
201930693 3146 39217 0901
201930699 3134 21633 0901

SÁNCHEZ SANTOS, LUIS FERNANDO
HERNANDEZ SAPÓN, LEVÍ ISAAC
MORALES XICARA, ERICK DANIEL

Índice

Índice.....	1
Introducción.....	2
Análisis de los resultados (Grupal).....	3
Caso de Aplicación.....	4
Bitácora.....	7
Creación de Backups (Actividades 1-5).....	7
Actividad 1.....	7
Actividad 2.....	14
Actividad 3.....	20
Actividad 4.....	28
Actividad 5.....	34
Restauración de Full backup (Actividades 6-10).....	42
Actividad 6.....	42
Actividad 7.....	49
Actividad 8.....	56
Actividad 9.....	62
Actividad 10.....	68
Restauración de backup incremental (Actividades 11-15).....	74
Actividad 11.....	74
Actividad 12.....	78
Actividad 13.....	82
Actividad 14.....	88
Actividad 15.....	93
Conclusiones.....	101

Introducción

En el ámbito de la ingeniería de sistemas, la gestión eficiente y segura de los datos es una responsabilidad fundamental, especialmente en entornos de producción donde se maneja un gran volumen de información. Los conceptos de backups y restauración son esenciales para garantizar la integridad y disponibilidad de los datos ante posibles fallos o pérdidas. En el curso de Sistemas de Bases de Datos 2, hemos profundizado en estas técnicas, preparando a los futuros DBA (Database Administrators) para enfrentar desafíos reales en el manejo y protección de datos.

El presente trabajo tiene como objetivo aplicar los conocimientos adquiridos en el curso, específicamente en la generación y restauración de backups en un entorno de producción. Para este ejercicio práctico, una clínica médica en Guatemala nos ha solicitado cargar sus datos almacenados en archivos de Excel a una base de datos, implementar estrategias de backup diarias y evaluar la efectividad de distintos tipos de backup: completo e incremental.

El proceso consistirá en realizar una carga masiva de datos utilizando un modelo entidad-relación proporcionado, seguido de la ejecución diaria de backups completos e incrementales. Al finalizar la carga, se eliminará la base de datos y se procederá a restaurar cada uno de los backups generados, registrando el tiempo que toma cada operación y comparando los resultados. Este análisis permitirá determinar el tipo de backup más adecuado para el volumen de datos manejado por la clínica.

Análisis de los resultados (Grupal)

La combinación de ambos tipos de backups (Full backup e incremental backup) nos permitió establecer una estrategia de protección de datos que fuera eficiente. Pudimos notar que mientras el full backup es crucial para tener una imagen completa de la base de datos, los backups incrementales son una mejor opción para operaciones diarias ya que estos suelen ocupar menos espacio lo que hace que se consuman menos recursos.

El tipo de restauración por medio de full backup demostró ser un proceso bastante robusto y confiable. A lo largo de cada una de las actividades relacionadas con la restauración por medio de full backup (Actividades 6 - 10) en las que debíamos eliminar los datos de nuestra base de datos, la base de datos volvía a su estado original sin ningún inconveniente. Esta práctica hizo que notemos la importancia de los full backups en la estrategia de recuperación ante desastres. No obstante, uno de los problemas principales en este tipo de backups es que toda la información es cargada en un solo documento. Esto da como resultado que, a diferencia de los backups incrementales, al momento de restaurar el backup este puede llevar mucho tiempo. Lo que da como resultado un consumo excesivo de recursos y tiempos de espera largos.

Por otra parte, el backup incremental demostró ser eficiente, ya que este solo se aplica sobre los datos “nuevos” que se ingresan en la base de datos desde el último backup. Esto da como resultado un ahorro en el tiempo a la hora de realizar la restauración. lo cual es crítico en un entorno de producción real. Sin embargo, es importante señalar que este tipo de backups debe realizarse en orden ya que si no se hace de esa manera pueden existir inconsistencias en nuestra base de datos.

Caso de Aplicación

Una clínica médica en Guatemala le solicita a usted que haga una carga de los datos que tienen en archivos de Excel a una base de datos y aplique sus conocimientos en backups y restauraciones para poder manipular los datos que ellos tienen sin que se pierdan ya que manejan grandes cantidades de datos. Se le solicita que utilice el siguiente modelo entidad - relación para crear la base de datos.

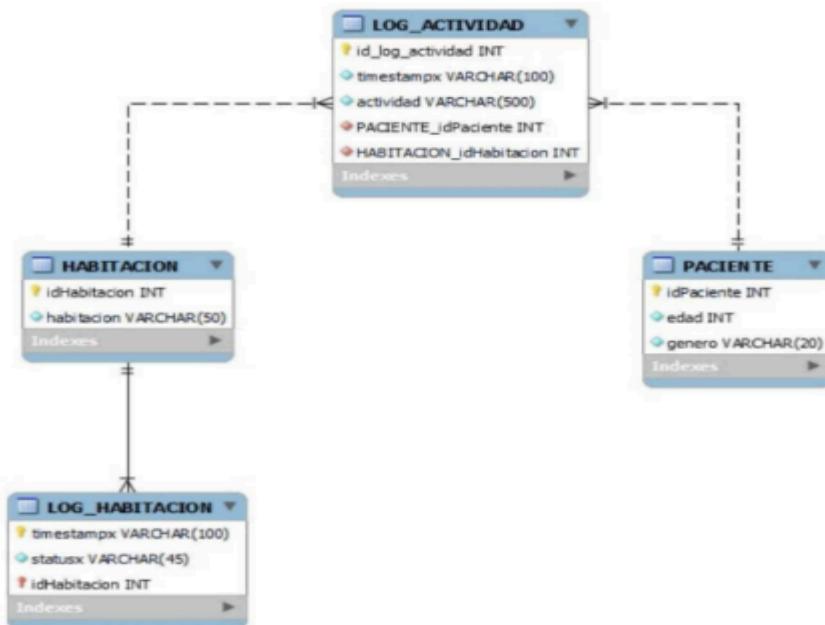
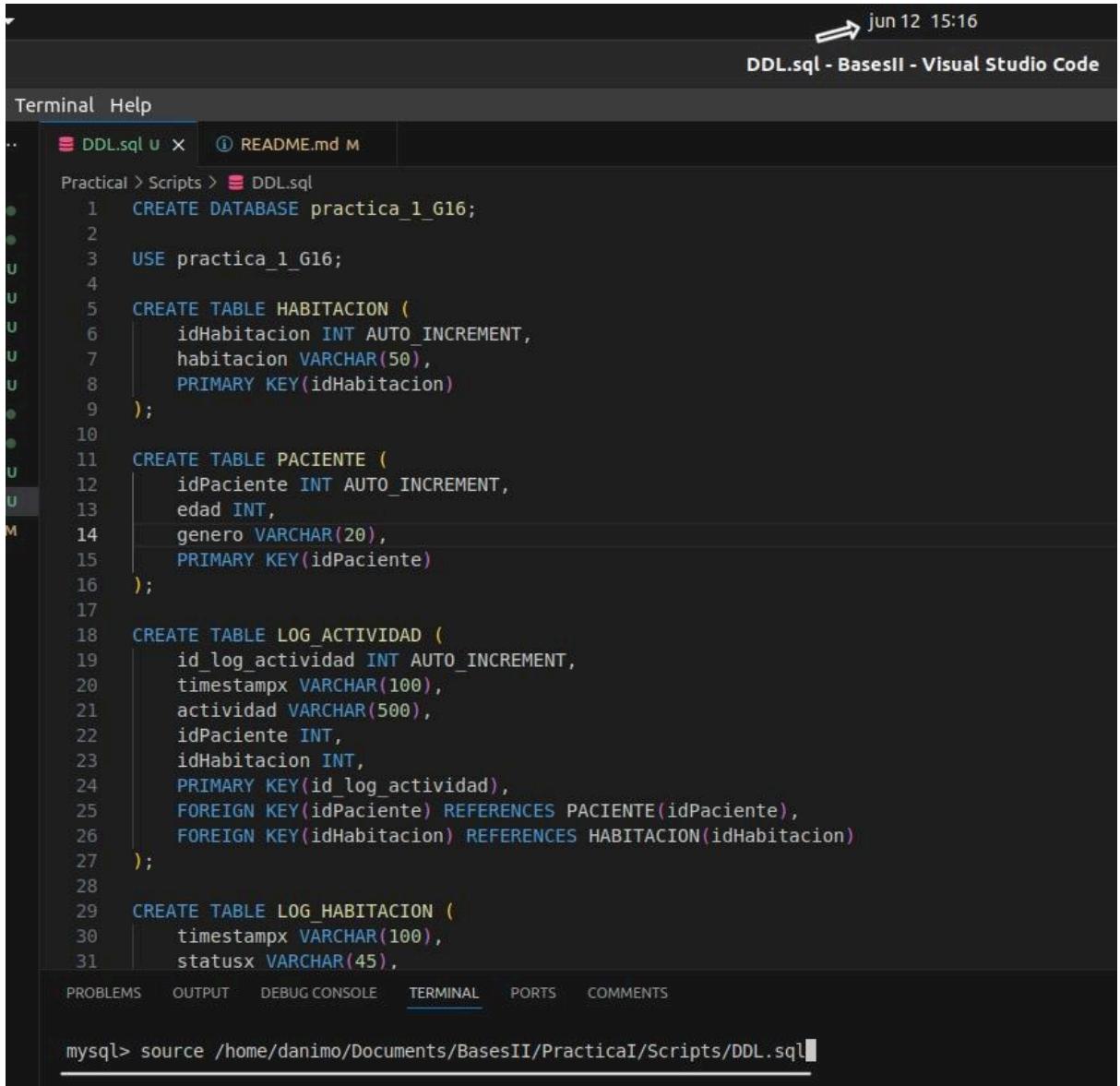


Imagen 0.1. Diagrama Relacional de la base de datos.

Para realizar la carga masiva de datos se le proporcionarán 5 archivos, los cuales usted deberá de cargar en la base de datos de la siguiente manera (Tomando en cuenta que somos un grupo par):

Día	Archivo
1	Habitaciones
2	Pacientes
3	LogActividades1
4	LogActividades2
5	LogHabitaciones

Dada esta propuesta para la base de datos. Primero creamos un script para la creación de esta en el sistema gestor de bases de datos MySQL.



The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal tab is selected at the bottom, showing the following MySQL DDL script:

```

CREATE DATABASE practica_1_G16;
USE practica_1_G16;
CREATE TABLE HABITACION (
    idHabitacion INT AUTO_INCREMENT,
    habitacion VARCHAR(50),
    PRIMARY KEY(idHabitacion)
);
CREATE TABLE PACIENTE (
    idPaciente INT AUTO_INCREMENT,
    edad INT,
    genero VARCHAR(20),
    PRIMARY KEY(idPaciente)
);
CREATE TABLE LOG_ACTIVIDAD (
    id_log_actividad INT AUTO_INCREMENT,
    timestampx VARCHAR(100),
    actividad VARCHAR(500),
    idPaciente INT,
    idHabitacion INT,
    PRIMARY KEY(id_log_actividad),
    FOREIGN KEY(idPaciente) REFERENCES PACIENTE(idPaciente),
    FOREIGN KEY(idHabitacion) REFERENCES HABITACION(idHabitacion)
);
CREATE TABLE LOG_HABITACION (
    timestampx VARCHAR(100),
    statusx VARCHAR(45),
);

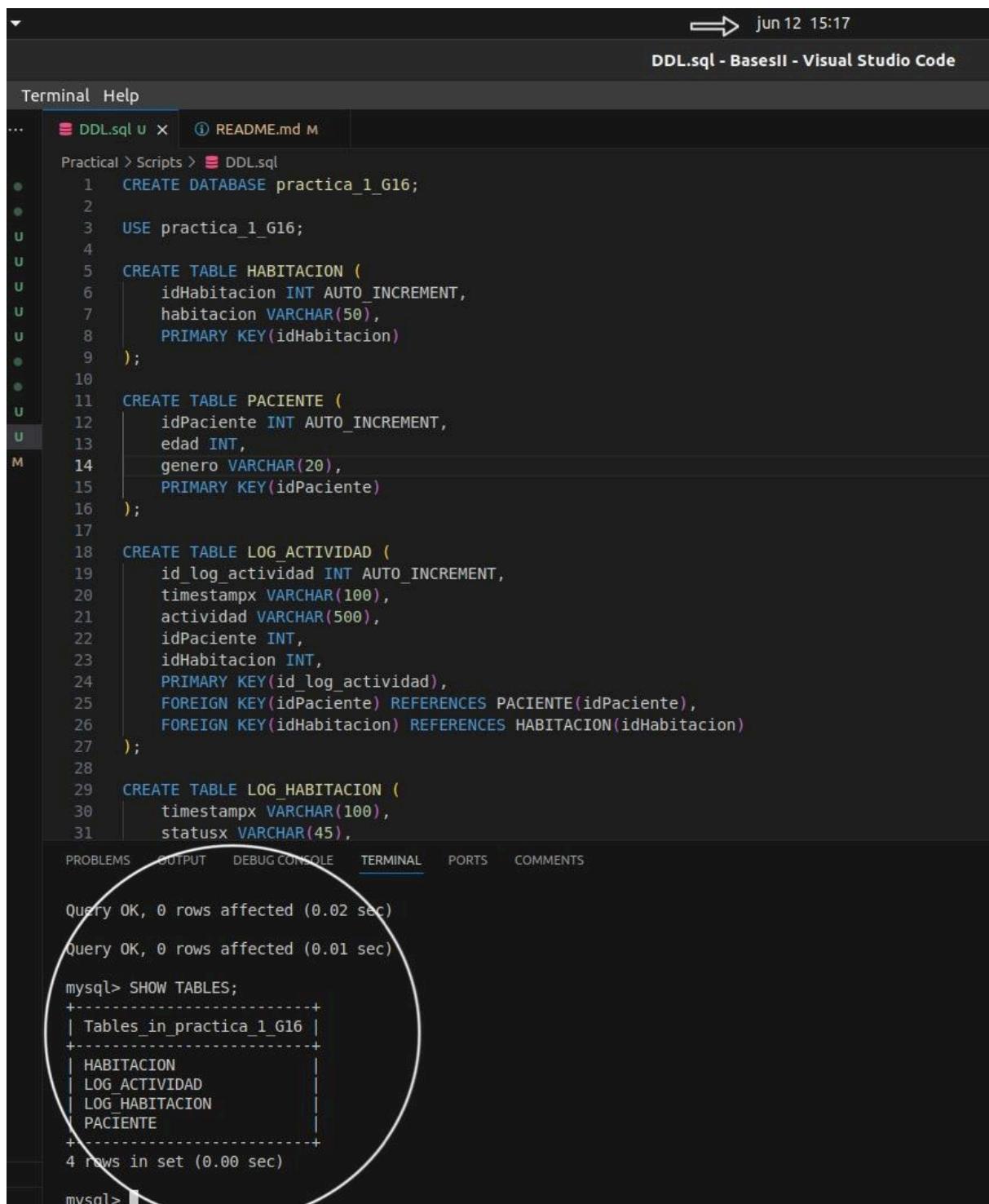
```

Below the script, the terminal shows the command:

```
mysql> source /home/danimo/Documents/BasesII/PracticaI/Scripts/DDL.sql
```

Imagen 0.2. Creación del script de la base de datos en MySQL.

Una vez creado el script que posee nuestra base de datos, procedemos a ejecutar dicho script con el comando source dentro de MySQL. Una vez hecho esto, corroboramos que se haya creado nuestra base de datos con las tablas correspondientes.



The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal shows the execution of a SQL script named DDL.sql, which creates a database and several tables. The terminal output indicates successful execution of each query. A circle highlights the MySQL command line interface where the user runs 'SHOW TABLES;' to verify the creation of the database and its tables.

```

Terminal Help
... DDL.sql u X README.md M
Practical > Scripts > DDL.sql
1 CREATE DATABASE practica_1_G16;
2
3 USE practica_1_G16;
4
5 CREATE TABLE HABITACION (
6     idHabitacion INT AUTO_INCREMENT,
7     habitacion VARCHAR(50),
8     PRIMARY KEY(idHabitacion)
9 );
10
11 CREATE TABLE PACIENTE (
12     idPaciente INT AUTO_INCREMENT,
13     edad INT,
14     genero VARCHAR(20),
15     PRIMARY KEY(idPaciente)
16 );
17
18 CREATE TABLE LOG_ACTIVIDAD (
19     id_log_actividad INT AUTO_INCREMENT,
20     timestampx VARCHAR(100),
21     actividad VARCHAR(500),
22     idPaciente INT,
23     idHabitacion INT,
24     PRIMARY KEY(id_log_actividad),
25     FOREIGN KEY(idPaciente) REFERENCES PACIENTE(idPaciente),
26     FOREIGN KEY(idHabitacion) REFERENCES HABITACION(idHabitacion)
27 );
28
29 CREATE TABLE LOG_HABITACION (
30     timestampx VARCHAR(100),
31     statusx VARCHAR(45),
32
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Query OK, 0 rows affected (0.02 sec)
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_practica_1_G16 |
+-----+
| HABITACION
| LOG_ACTIVIDAD
| LOG_HABITACION
| PACIENTE
+-----+
4 rows in set (0.00 sec)

mysql>

```

Imagen 0.3. Confirmación de creación de Base de datos.

Ahora que se ha creado nuestra base de datos, podemos dar inicio a la realización de todas las actividades propuestas en la práctica.

Bitácora

Creación de Backups (Actividades 1-5)

Actividad 1

Para la primera actividad lo primero que se debe hacer es la carga de datos para la tabla **HABITACION**. Esto lo hacemos con la ayuda de Workbench haciendo uso de la importación de datos Wizard.

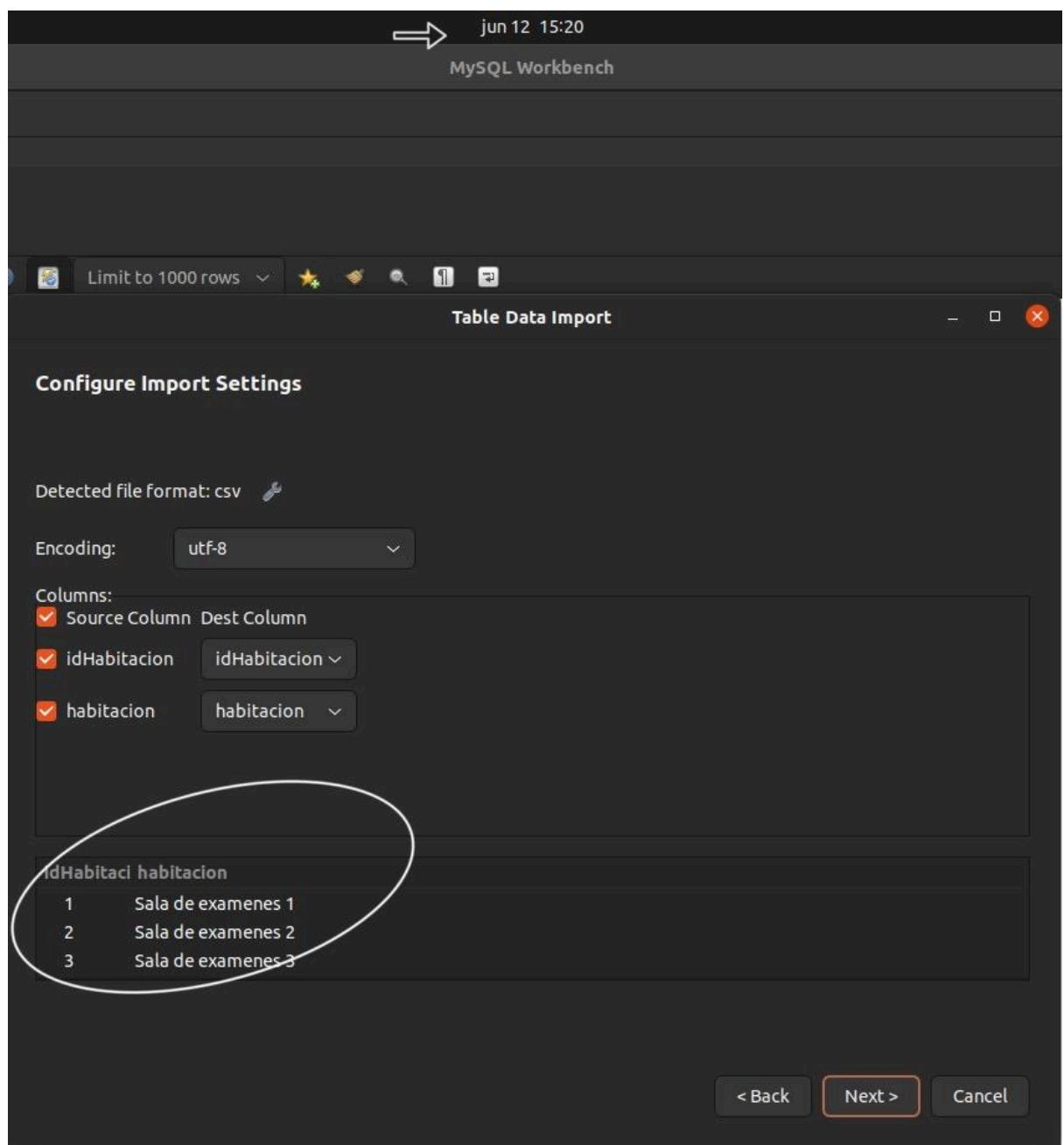


Imagen 1.1. Carga de datos en la tabla HABITACION.

Ahora que nuestra tabla posee información. El siguiente paso es ejecutar una instrucción SELECT y una instrucción COUNT para corroborar que todos los datos se hayan cargado de forma correcta. Primero mostraremos los resultados obtenidos con el SELECT a cada tabla.

The screenshot shows the MySQL Workbench interface. In the top right corner, it says "jun 12 15:22". The main window has a title bar "Query 1" and a toolbar with various icons. Below the toolbar, there is a code editor containing the following SQL script:

```
1 • USE practica_1_G16;
2
3 • SELECT * FROM HABITACION;
```

The third line, "SELECT * FROM HABITACION;", is highlighted with a blue selection bar. Below the code editor is a results grid. The first few rows of the grid are:

#	idHabitacion	habitacion
1	1	Sala de examenes 1
2	2	Sala de examenes 2
3	3	Sala de examenes 3
4	4	Sala de examenes 4
5	5	Sala de imagenes 1
6	6	Sala de procedimientos 1
7	7	Sala de procedimientos 2
8	8	Sala de procedimientos 3
9	9	Sala de procedimientos 4
10	10	Repcion
11	11	Laboratorio
12	12	Estacion de revision 1
13	13	Estacion de revision 2
14	14	Estacion de revision 3
15	15	Estacion de revision 4
	NUL	HULL

Imagen 1.2. SELECT de la tabla HABITACION.

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query editor contains the following SQL code:

```
1 • USE practica_1_G16;
2
3 • SELECT * FROM HABITACION;
4 • SELECT * FROM PACIENTE;
5 • SELECT * FROM LOG_ACTIVIDAD;
6 • SELECT * FROM LOG_HABITACION;
```

The results grid displays the following data:

#	idPaciente	edad	genero
1	NULL	NULL	NULL

Imagen 1.3. SELECT de la tabla PACIENTE.

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query editor contains the following SQL code:

```
1 • USE practica_1_G16;
2
3 • SELECT * FROM HABITACION;
4 • SELECT * FROM PACIENTE;
5 • SELECT * FROM LOG_ACTIVIDAD;
6 • SELECT * FROM LOG_HABITACION;
```

The results grid displays the following data:

#	id_log_actividad	timestamp	actividad	idPaciente	idHabitacion
1	NULL	NULL	NULL	NULL	NULL

Imagen 1.4. SELECT de la tabla LOG_ACTIVIDAD.

The screenshot shows the MySQL Workbench interface. In the top right corner, it says "jun 12 15:23" and "MySQL Workbench". The main area is titled "Query 1". The code entered is:

```

1 • USE practica_1_G16;
2
3 • SELECT * FROM HABITACION;
4 • SELECT * FROM PACIENTE;
5 • SELECT * FROM LOG_ACTIVIDAD;
6 • SELECT * FROM LOG_HABITACION;

```

Below the code, the results are displayed in a table with columns: #, timestamp, status, idHabitacion. The first row shows:

#	timestamp	status	idHabitacion
1	NULL	HULL	NULL

Imagen 1.5. SELECT de la tabla LOG_HABITACION.

Ahora mostraremos los resultados obtenidos con el comando SELECT COUNT(*) a cada tabla de nuestra base de datos.

The screenshot shows the MySQL Workbench interface. In the top right corner, it says "jun 12 15:27" and "MySQL Workbench". The main area is titled "Query 1". The code entered is:

```

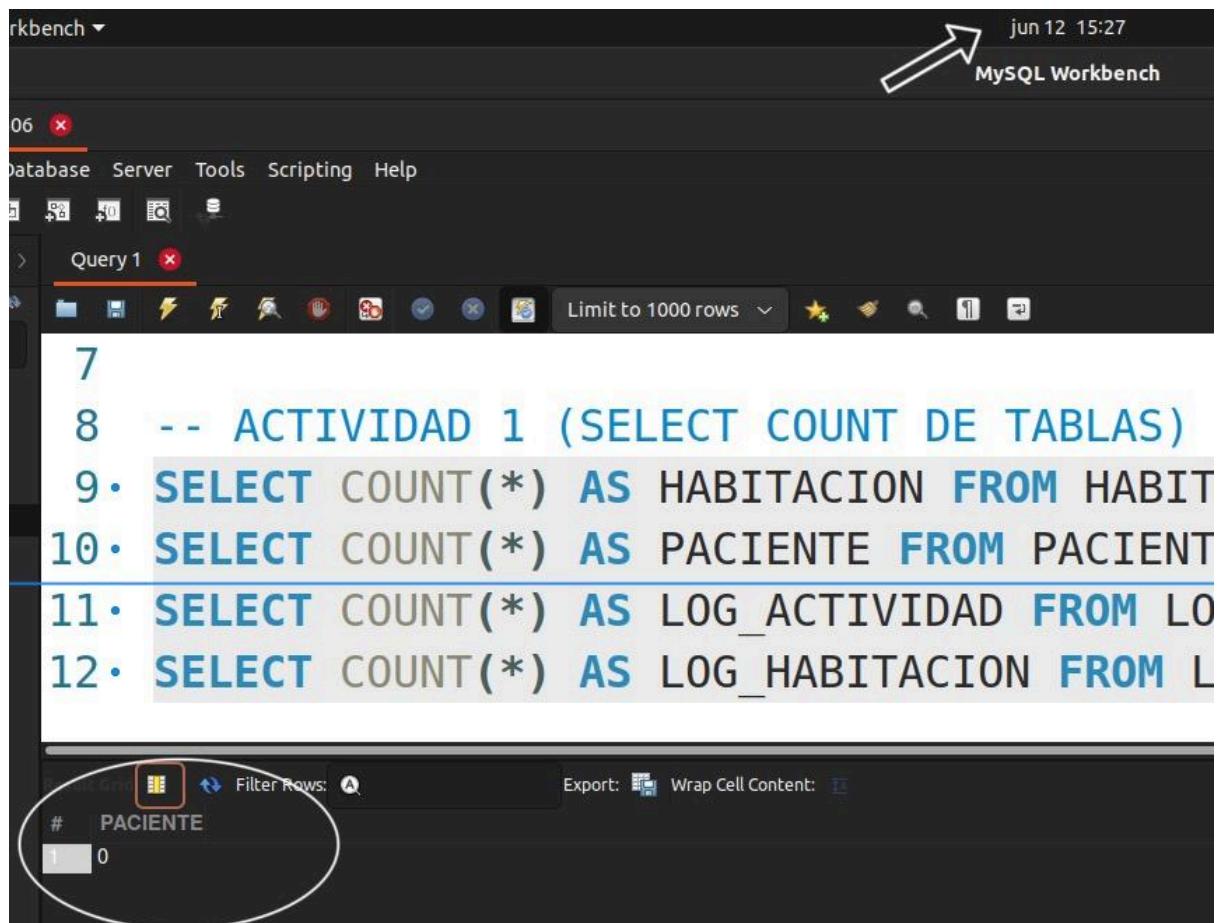
7
8 -- ACTIVIDAD 1 (SELECT COUNT DE TABLAS)
9 • SELECT COUNT(*) AS HABITACION FROM HABITACION
10• SELECT COUNT(*) AS PACIENTE FROM PACIENTE
11• SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD
12• SELECT COUNT(*) AS LOG_HABITACION FROM LOG_HABITACION

```

Below the code, the results are displayed in a table with columns: #, HABITACION. The first row shows:

#	HABITACION
1	15

Imagen 1.6. SELECT COUNT(*) a la tabla HABITACION.

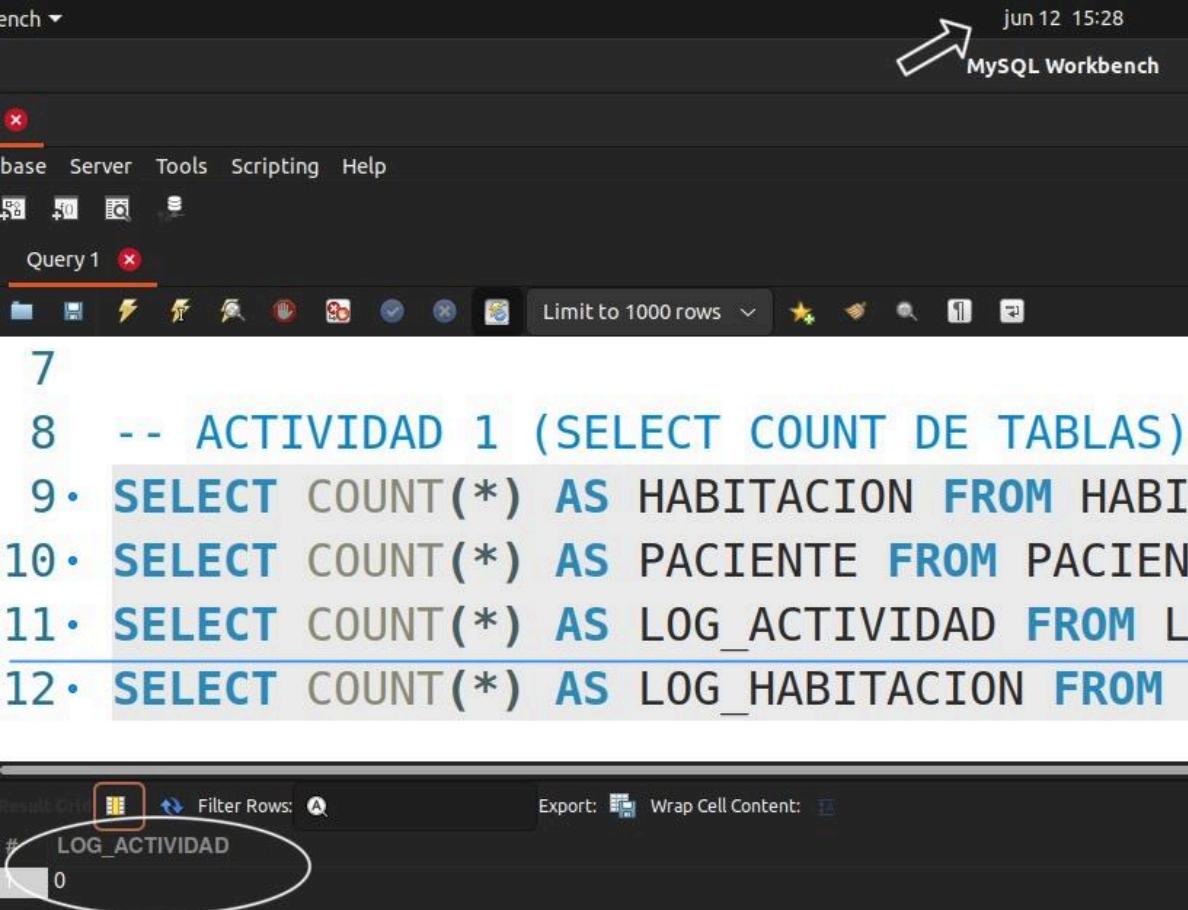


The screenshot shows the MySQL Workbench interface. The title bar reads "jun 12 15:27 MySQL Workbench". The menu bar includes "Database", "Server", "Tools", "Scripting", and "Help". The toolbar has various icons for database management. A query editor window titled "Query 1" contains the following SQL code:

```
7  
8 -- ACTIVIDAD 1 (SELECT COUNT DE TABLAS)  
9 • SELECT COUNT(*) AS HABITACION FROM HABITACION  
10 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE  
11 • SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD  
12 • SELECT COUNT(*) AS LOG_HABITACION FROM LOG_HABITACION
```

Below the query editor is a results grid for the "PACIENTE" table. The grid has columns "#", "PACIENTE", and "0". The "#" column is highlighted with a red oval. The "PACIENTE" column header is also highlighted with a red oval.

Imagen 1.7. SELECT COUNT(*) a la tabla PACIENTE.



The screenshot shows the MySQL Workbench interface. The top bar displays the date and time as 'jun 12 15:28' and the application name 'MySQL Workbench'. The menu bar includes 'Database', 'Server', 'Tools', 'Scripting', and 'Help'. Below the menu is a toolbar with various icons. A tab labeled 'Query 1' is active. The main area contains the following SQL code:

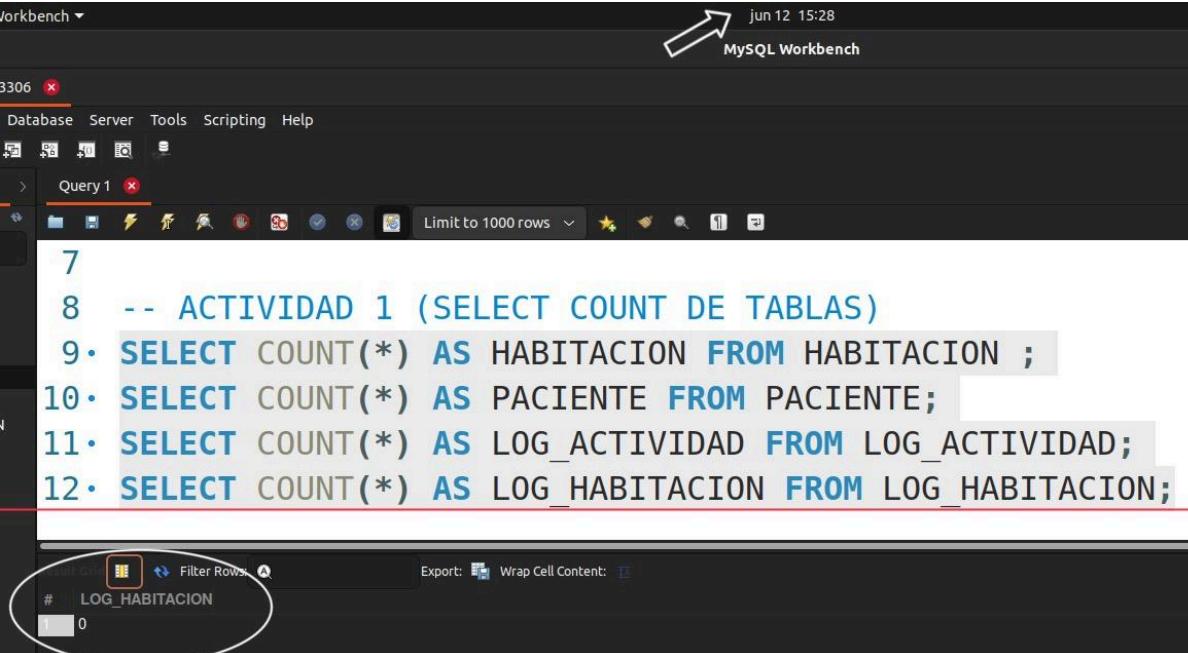
```

7
8 -- ACTIVIDAD 1 (SELECT COUNT DE TABLAS)
9 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
10 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE ;
11 • SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD ;
12 • SELECT COUNT(*) AS LOG_HABITACION FROM LOG_HABITACION ;

```

The result grid shows a single row for the 'LOG_ACTIVIDAD' table, with the value '0' in the first column. The result grid has a header row with columns for '#', 'LOG_ACTIVIDAD', and '0'.

Imagen 1.8. SELECT COUNT(*) a la tabla LOG_ACTIVIDAD.



The screenshot shows the MySQL Workbench interface. The top bar displays the date and time as 'jun 12 15:28' and the application name 'MySQL Workbench'. The menu bar includes 'Database', 'Server', 'Tools', 'Scripting', and 'Help'. Below the menu is a toolbar with various icons. A tab labeled 'Query 1' is active. The main area contains the following SQL code:

```

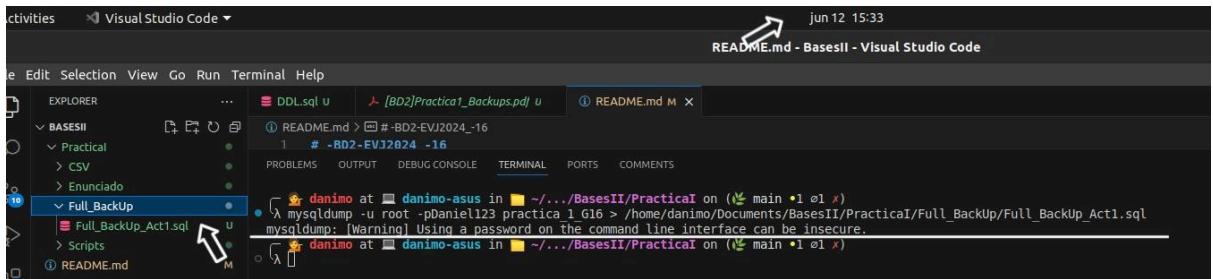
7
8 -- ACTIVIDAD 1 (SELECT COUNT DE TABLAS)
9 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
10 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE ;
11 • SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD ;
12 • SELECT COUNT(*) AS LOG_HABITACION FROM LOG_HABITACION ;

```

The result grid shows a single row for the 'LOG_HABITACION' table, with the value '0' in the first column. The result grid has a header row with columns for '#', 'LOG_HABITACION', and '0'.

Imagen 1.9. SELECT COUNT(*) a la tabla LOG_HABITACION.

Una vez verificada la carga correcta de los datos. Procedemos a crear los backups correspondientes a la primera actividad. Primero crearemos el Full Backup de nuestra base de datos.



```
jun 12 15:33
README.md - BasesII - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER README.md
BASESII DDL.sql u [BD2]Practica1_Backups.pdf u
    Practical ...
        CSV ...
        Enunciado ...
    Full_BackUp Full_BackUp_Act1.sql Scripts README.md
# -BD2-EVJ2024 -16
1 mysqldump -u root -pDaniell23 practica1 G16 > /home/danimo/Documents/BasesII/PracticaI/Full_BackUp/Full_BackUp_Act1.sql
mysqldump: [Warning] Using a password on the command line interface can be insecure.
danimo at danimo-asus in ~/.../BasesII/Practica1 on (main • 1 ⏺)
```

Imagen 1.10. Creación de Full Backup de nuestra base de datos.

Luego de crear nuestro Full Backup, creamos nuestro backup incremental sobre la tabla HABITACION.



```
jun 12 15:38
README.md - BasesII - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER README.md
BASESII DDL.sql u
    Practical ...
        BackUp_Incremental Incremental_Backup_Act...
            CSV ...
            Enunciado ...
        Scripts README.md
# -BD2-EVJ2024 -16
1 mysqldump -u root -pDaniell23 practica1 HABITACION > /home/danimo/Documents/BasesII/PracticaI/BackUp_Incremental/Incremental_Backup_Act1.sql
mysqldump: [Warning] Using a password on the command line interface can be insecure.
danimo at danimo-asus in ~/.../BasesII/Practica1 on (main • 1 ⏺)
```

Imagen 1.11. Creación de backup incremental de la tabla HABITACION.

Actividad 2

Para la segunda actividad comenzamos cargando los datos para la tabla PACIENTE. Al igual que en la primera actividad, hicimos uso de la herramienta wizard en workbench.

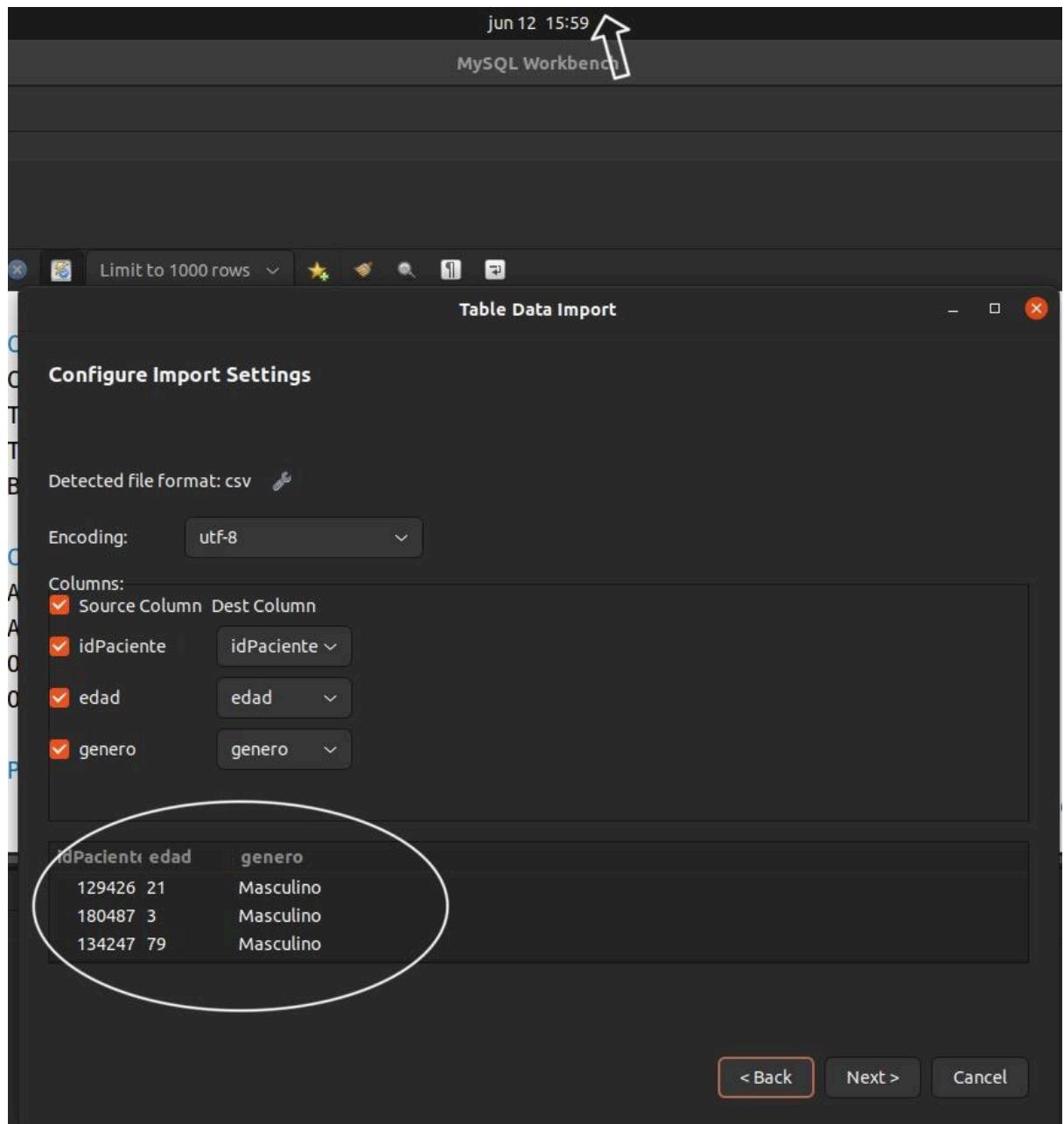


Imagen 2.1. Carga de datos a la tabla PACIENTE.

Una vez finalizada la carga de datos, procedemos a ejecutar comandos SELECT para verificar que la carga de datos se haya realizado correctamente.

The screenshot shows the MySQL Workbench interface with a dark theme. At the top, there's a toolbar with various icons. Below it is a menu bar with 'Database', 'Server', 'Tools', 'Scripting', and 'Help'. A status bar at the top right indicates 'jun 12 19:00' and 'MySQL Workbench'. The main area is titled 'Query 1' and contains the following SQL code:

```
3 •  SELECT * FROM HABITACION;
4 •  SELECT * FROM PACIENTE;
5 •  SELECT * FROM LOG_ACTIVIDAD;
6 •  SELECT * FROM LOG_HABITACION;
```

Below the code, the 'Result Grid' tab is selected, showing the results of the first query. The results are as follows:

#	idHabitacion	habitacion
1	1	Sala de examenes 1
2	2	Sala de examenes 2
3	3	Sala de examenes 3
4	4	Sala de examenes 4
5	5	Sala de imagenes 1
6	6	Sala de procedimientos 1
7	7	Sala de procedimientos 2
8	8	Sala de procedimientos 3
9	9	Sala de procedimientos 4
10	10	Recepcion
11	11	Laboratorio
12	12	Estación de revisión 1
13	13	Estación de revisión 2
14	14	Estación de revisión 3
15	15	Estación de revisión 4
*	NULL	NULL

Imagen 2.2. SELECT de la tabla HABITACION.

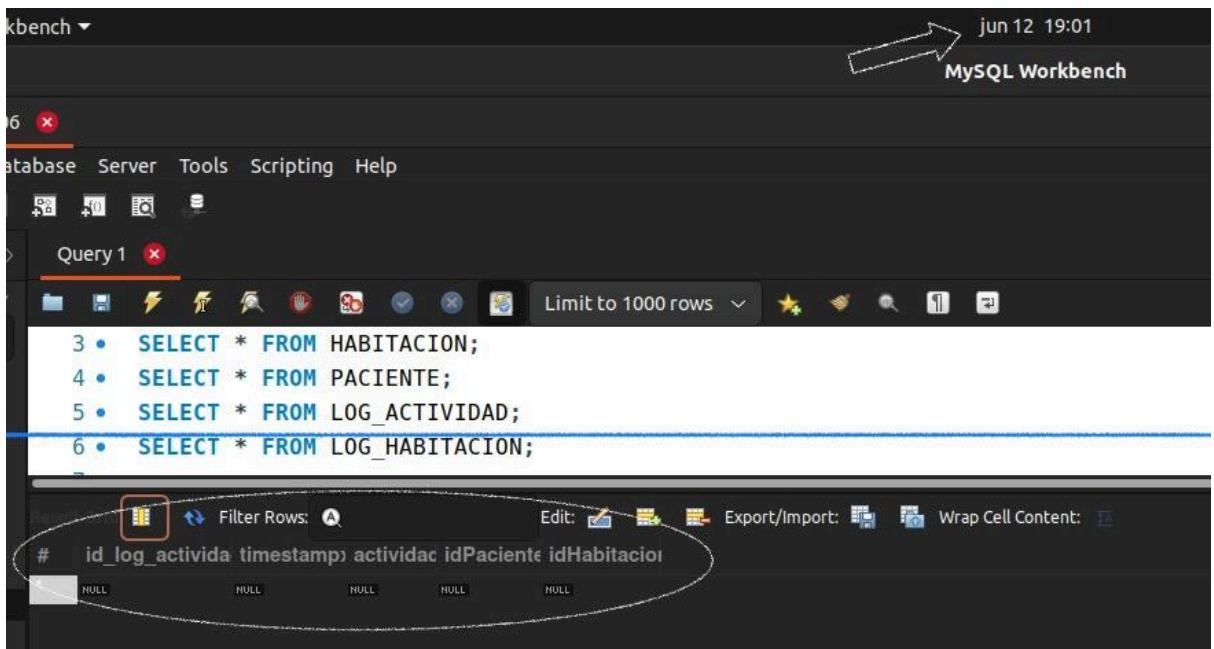
The screenshot shows the MySQL Workbench interface. At the top, it displays the date and time as "jun 12 19:01" and the title "MySQL Workbench". The main area shows a query editor titled "Query 1" containing the following SQL code:

```
3 •  SELECT * FROM HABITACION;
4 •  SELECT * FROM PACIENTE;
5 •  SELECT * FROM LOG_ACTIVIDAD;
6 •  SELECT * FROM LOG_HABITACION;
```

Below the code, the "Result Grid" tab is selected, showing the results of the second query (SELECT * FROM PACIENTE). The results are presented in a table with columns: #, idPaciente, edad, and genero. The data consists of 28 rows, each representing a patient record. The "HABITACION 25", "PACIENTE 26", "LOG_ACTIVIDAD 27", and "LOG_HABITACION 28" tabs are also visible at the bottom.

#	idPaciente	edad	genero
1	100000	95	Otro
2	100001	40	Femenino
3	100002	42	Masculino
4	100003	8	Femenino
5	100004	88	Masculino
6	100005	23	Masculino
7	100006	60	Femenino
8	100007	49	Femenino
9	100008	44	Femenino
10	100009	55	Femenino
11	100010	28	Femenino
12	100011	13	Femenino
13	100012	62	Masculino
14	100013	32	Masculino
15	100014	11	Masculino
16	100015	50	Femenino
17	100016	39	Masculino
18	100017	55	Femenino
19	100018	61	Masculino
20	100019	58	Masculino
21	100020	6	Masculino
22	100021	26	Femenino
23	100022	2	Masculino
24	100023	6	Femenino
25	100024	97	Masculino
26	100025	67	Femenino
27	100026	57	Femenino
28	100027	54	Femenino

Imagen 2.3. SELECT de la tabla PACIENTE.



The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query executed is:

```

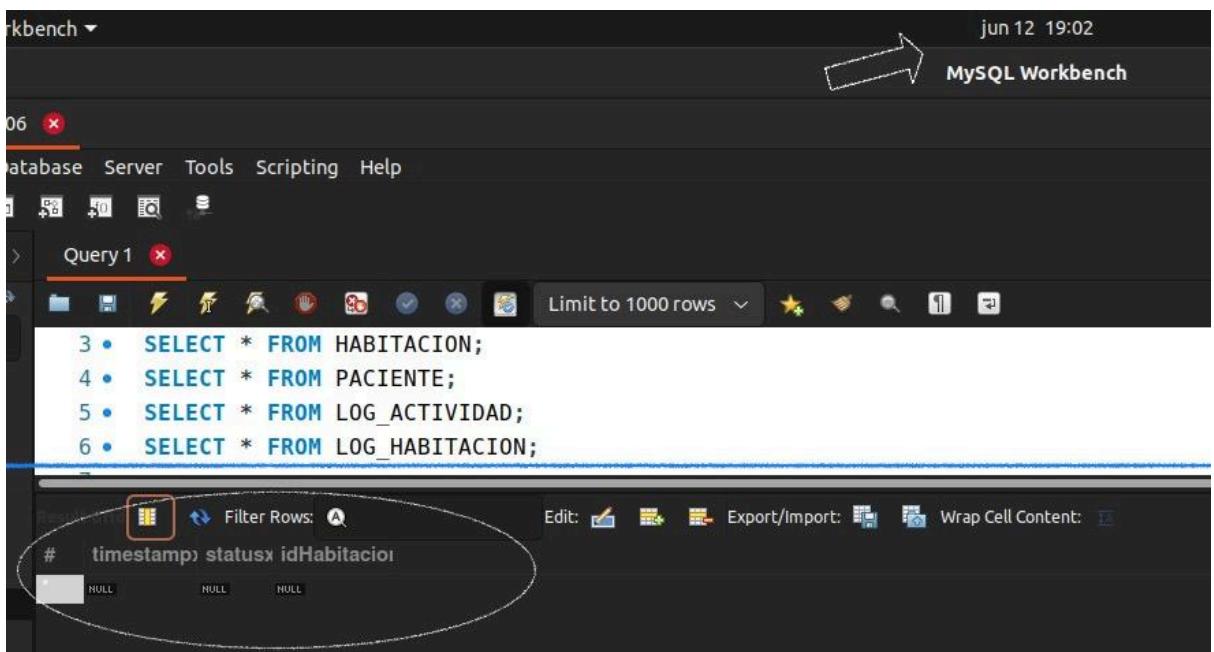
3 •  SELECT * FROM HABITACION;
4 •  SELECT * FROM PACIENTE;
5 •  SELECT * FROM LOG_ACTIVIDAD;
6 •  SELECT * FROM LOG_HABITACION;

```

The results pane displays the structure of the LOG_ACTIVIDAD table:

#	id_log_actividad	timestamp	actividad	idPaciente	idHabitacion
1	NULL	NULL	NULL	NULL	NULL

Imagen 2.4. SELECT de la tabla LOG_ACTIVIDAD.



The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query executed is:

```

3 •  SELECT * FROM HABITACION;
4 •  SELECT * FROM PACIENTE;
5 •  SELECT * FROM LOG_ACTIVIDAD;
6 •  SELECT * FROM LOG_HABITACION;

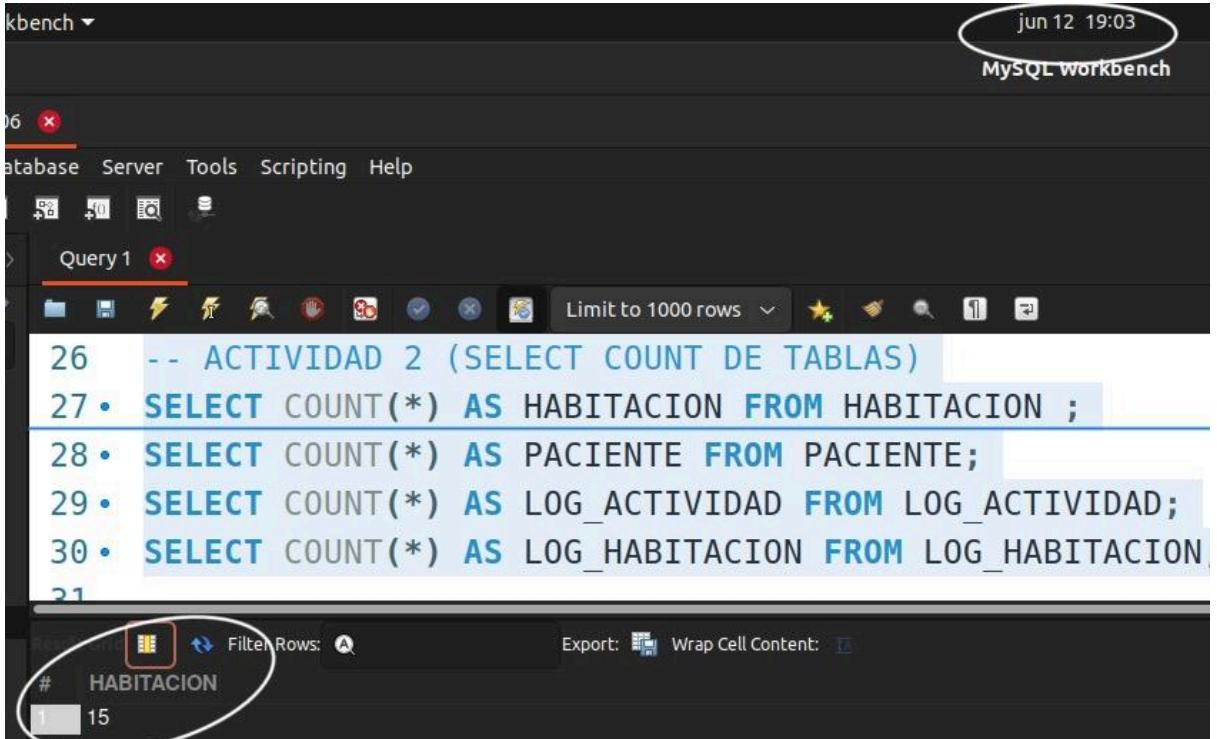
```

The results pane displays the structure of the LOG_HABITACION table:

#	timestamp	status	idHabitacion
1	HULL	HULL	HULL

Imagen 2.5. SELECT de la tabla LOG_HABITACION.

También ejecutamos comandos SELECT COUNT(*) a cada tabla para corroborar que cada tabla cuenta con todos los datos cargados.



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

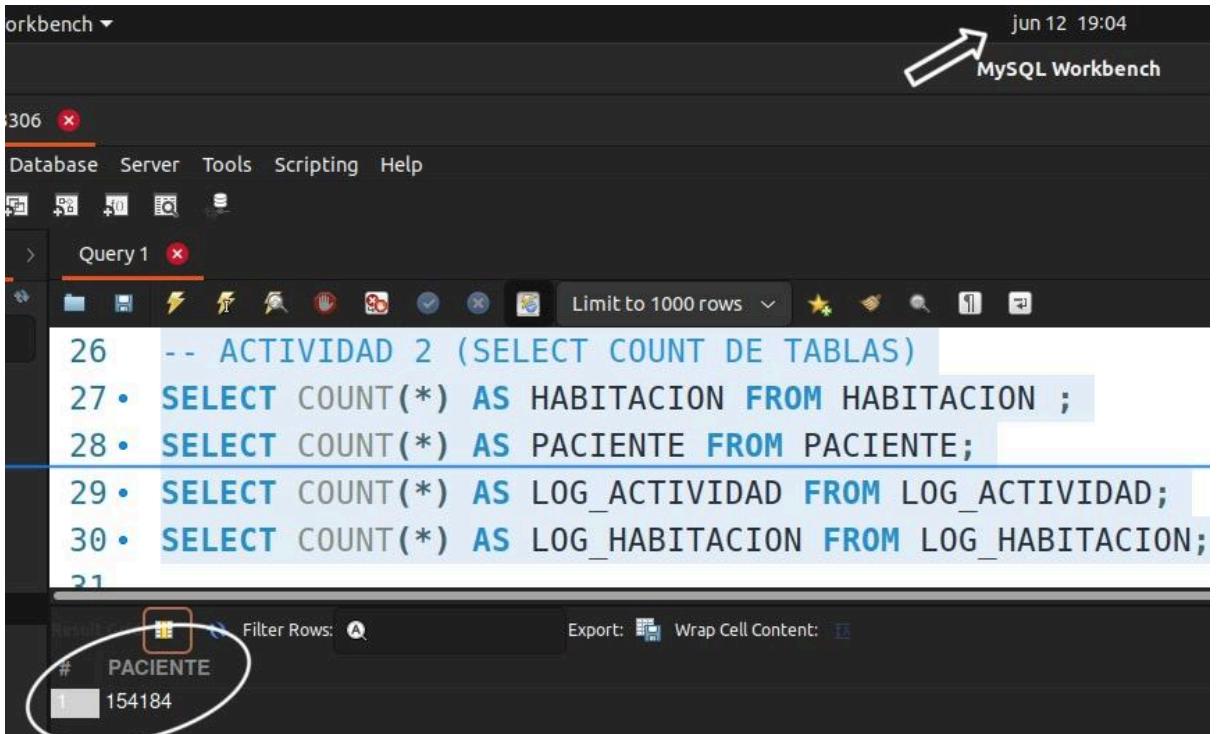
26 -- ACTIVIDAD 2 (SELECT COUNT DE TABLAS)
27 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
28 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE;
29 • SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD;
30 • SELECT COUNT(*) AS LOG_HABITACION FROM LOG_HABITACION;
21

```

The result set for the first query (HABITACION) is displayed below the table header:

#	HABITACION
1	15

Imagen 2.6. SELECT COUNT(*) a la tabla HABITACION.



The screenshot shows the MySQL Workbench interface with a query editor containing the same SQL code as in Image 2.6:

```

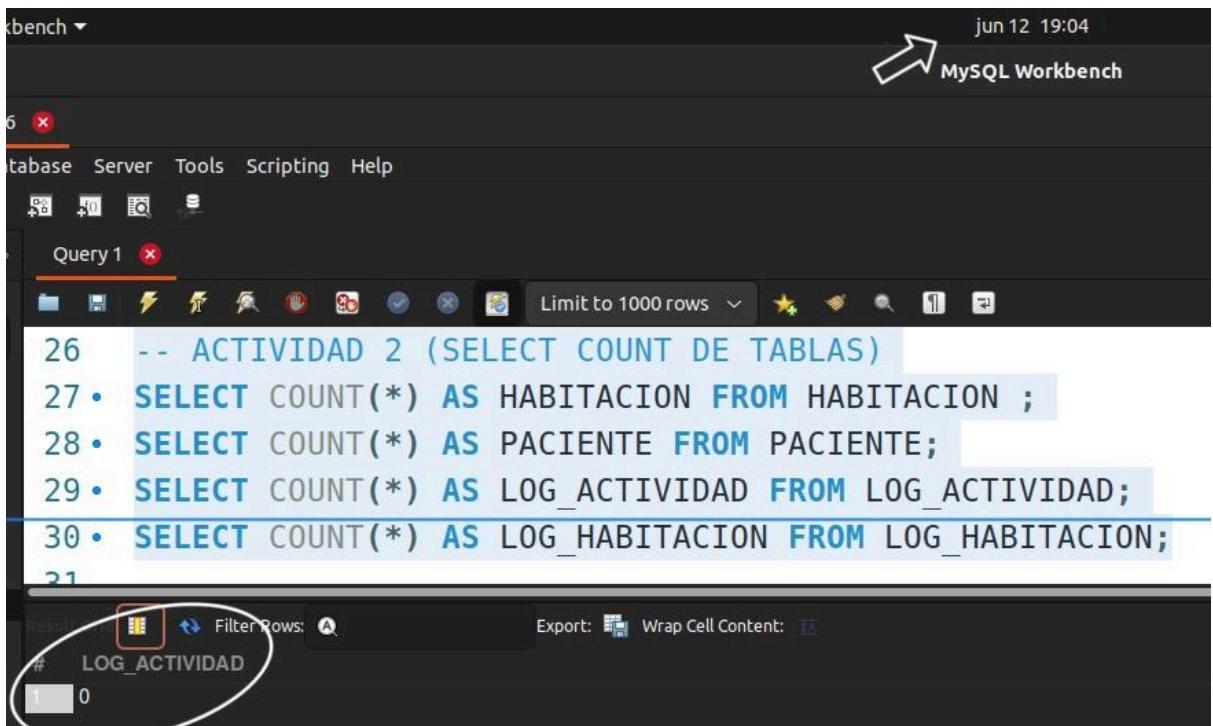
26 -- ACTIVIDAD 2 (SELECT COUNT DE TABLAS)
27 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
28 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE;
29 • SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD;
30 • SELECT COUNT(*) AS LOG_HABITACION FROM LOG_HABITACION;
21

```

The result set for the second query (PACIENTE) is displayed below the table header:

#	PACIENTE
1	154184

Imagen 2.7. SELECT COUNT(*) a la tabla PACIENTE.



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

26 -- ACTIVIDAD 2 (SELECT COUNT DE TABLAS)
27 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
28 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE;
29 • SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD;
30 • SELECT COUNT(*) AS LOG_HABITACION FROM LOG_HABITACION;
21

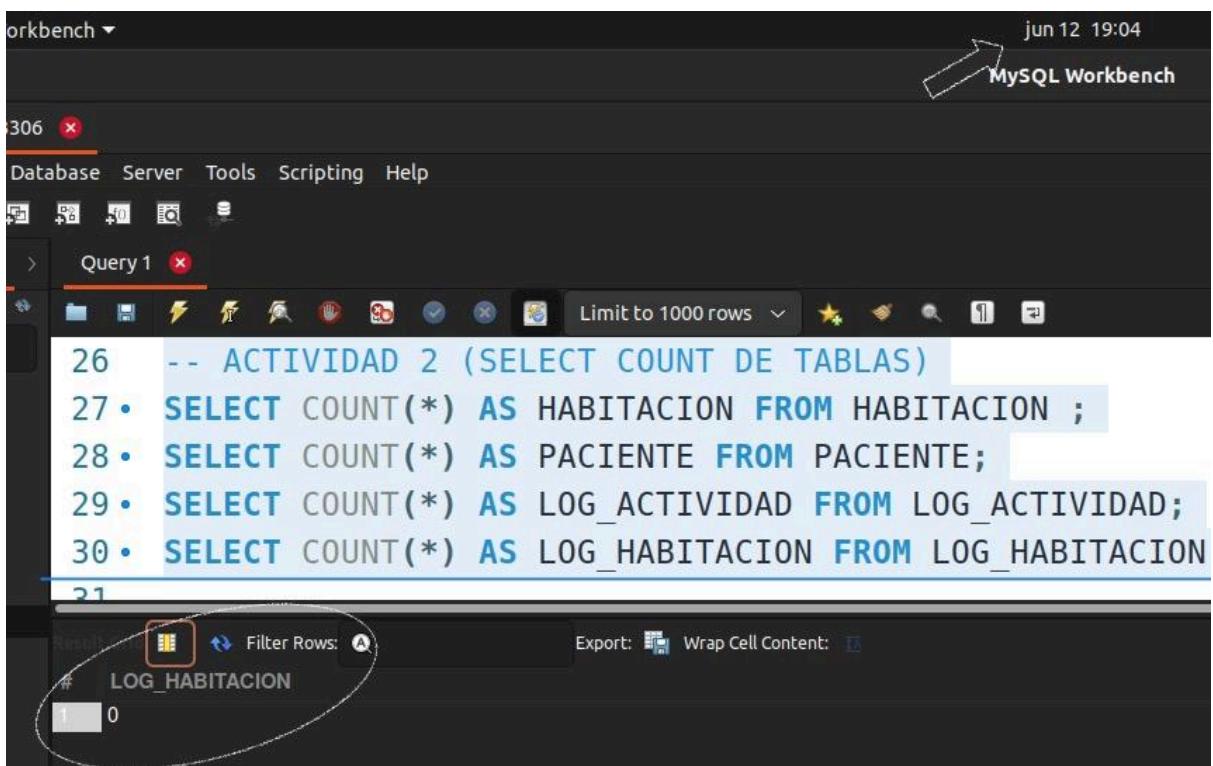
```

The results pane below the query editor displays the following table:

#	LOG_ACTIVIDAD
1	0

A red oval highlights the table header and the first row of the results table.

Imagen 2.8. SELECT COUNT(*) a la tabla LOG_ACTIVIDAD.



The screenshot shows the MySQL Workbench interface with a query editor containing the same SQL code as in Image 2.8:

```

26 -- ACTIVIDAD 2 (SELECT COUNT DE TABLAS)
27 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
28 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE;
29 • SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD;
30 • SELECT COUNT(*) AS LOG_HABITACION FROM LOG_HABITACION;
21

```

The results pane displays the following table:

#	LOG_HABITACION
1	0

A red oval highlights the table header and the first row of the results table.

Imagen 2.9. SELECT COUNT(*) a la tabla LOG_HABITACION.

Ahora que sabemos que todos los datos se encuentran en nuestra base de datos podemos crear las copias de seguridad respectivas a la segunda actividad. Primero comenzaremos creando un full backup de la base de datos.

```
CREATE DATABASE práctica1 G16;
mysqldump: [Warning] Using a password on the command line interface can be insecure.
```

Imagen 2.10. Creación de full backup de nuestra base de datos.

Luego de crear el full backup, creamos el backup incremental para la tabla PACIENTE.

```
mysqldump: [Warning] Using a password on the command line interface can be insecure.
```

Imagen 2.11. Creación de backup incremental para la tabla PACIENTE.

Actividad 3

Para realizar la actividad número tres comenzamos cargando los datos del archivo csv LogActividades1 a nuestra tabla LOG_ACTIVIDAD.

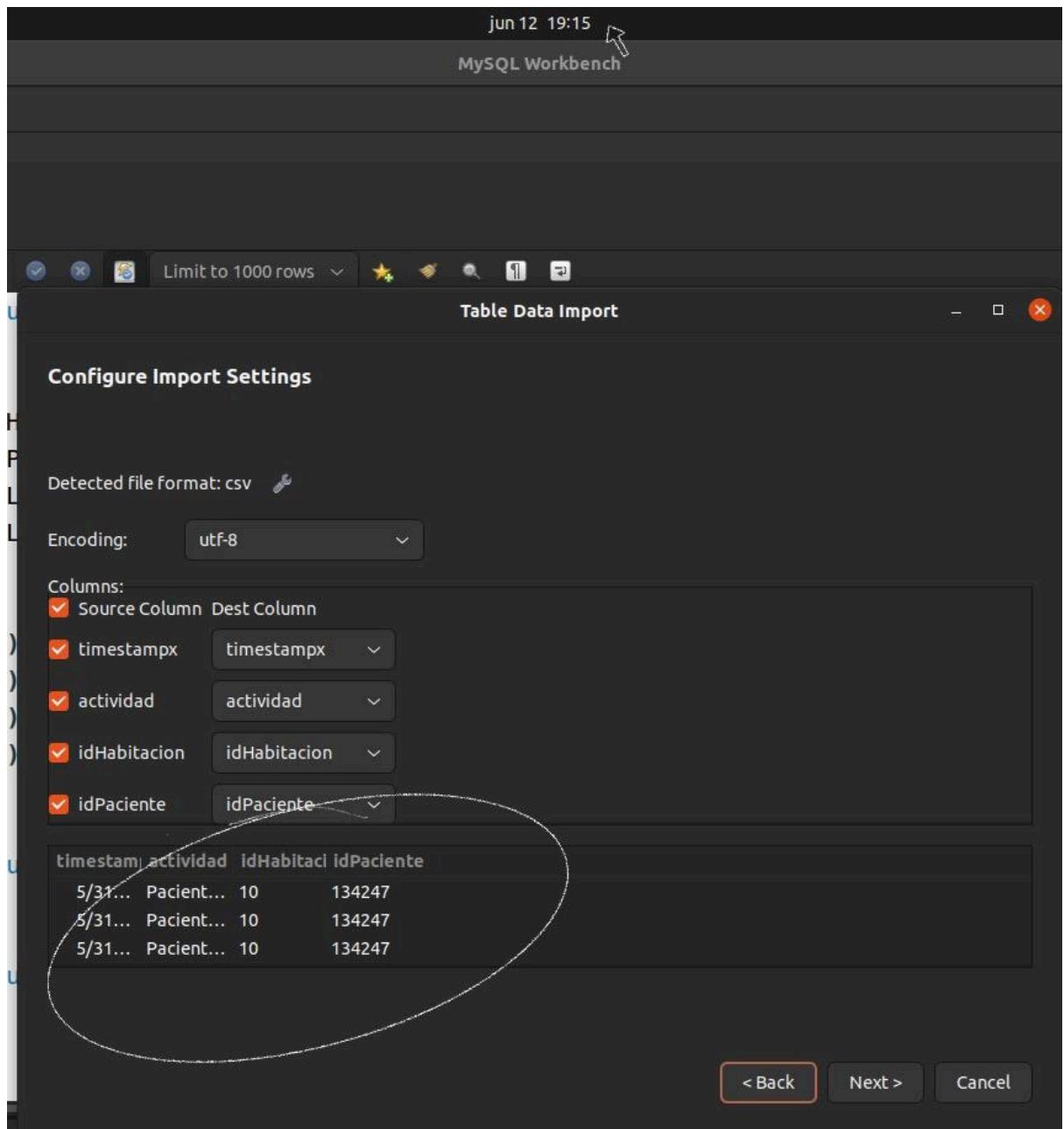


Imagen 3.1. Carga de datos a la tabla LOG_ACTIVIDAD.

Una vez realizada la carga de datos, procedemos a ejecutar el comando SELECT en cada una de las tablas para verificar que los datos han sido cargados correctamente a nuestra base de datos.

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a menu bar with 'File', 'Edit', 'Server', 'Tools', 'Scripting', and 'Help'. A status bar at the top right indicates 'jun 12 19:22' and 'MySQL Workbench'. The main area has a tab labeled 'Query 1' with a red close button. Below the tabs is a toolbar with icons for file operations, search, and execution. A 'Limit to 1000 rows' dropdown is also present. The query editor contains the following code:

```
39 -- ACTIVIDAD 3 (SELECT DE TABLAS)
40 • SELECT * FROM HABITACION;
41 • SELECT * FROM PACIENTE;
42 • SELECT * FROM LOG_ACTIVIDAD;
43 • SELECT * FROM LOG_HABITACION;
```

Below the code is a result grid titled 'Result Grid'. It has two columns: '# idHabitacion' and 'habitacion'. The data is as follows:

#	idHabitacion	habitacion
1	1	Sala de examenes 1
2	2	Sala de examenes 2
3	3	Sala de examenes 3
4	4	Sala de examenes 4
5	5	Sala de imagenes 1
6	6	Sala de procedimientos 1
7	7	Sala de procedimientos 2
8	8	Sala de procedimientos 3
9	9	Sala de procedimientos 4
10	10	Recepcion
11	11	Laboratorio
12	12	Estación de revisión 1
13	13	Estación de revisión 2
14	14	Estación de revisión 3
15	15	Estación de revisión 4
*	NULL	NULL

A large white circle is drawn over the result grid, highlighting the data.

Imagen 3.2. SELECT a la tabla HABITACION.

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a menu bar with 'Base', 'Server', 'Tools', 'Scripting', and 'Help'. A status bar at the top right indicates 'jun 12 19:22' and 'MySQL Workbench'. The main area is titled 'Query 1' and contains the following SQL code:

```
39 -- ACTIVIDAD 3 (SELECT DE TABLAS)
40 • SELECT * FROM HABITACION;
41 • SELECT * FROM PACIENTE;
42 • SELECT * FROM LOG_ACTIVIDAD;
43 • SELECT * FROM LOG_HABITACION;
```

Below the code is a result grid titled 'Result Grid' with a 'Filter Rows' button. The grid displays data from the 'PACIENTE' table:

#	idPaciente	edad	genero
1	100000	95	Otro
2	100001	40	Femenino
3	100002	42	Masculino
4	100003	8	Femenino
5	100004	88	Masculino
6	100005	23	Masculino
7	100006	60	Femenino
8	100007	49	Femenino
9	100008	44	Femenino
10	100009	55	Femenino
11	100010	28	Femenino
12	100011	13	Femenino
13	100012	62	Masculino
14	100013	32	Masculino
15	100014	11	Masculino
16	100015	50	Femenino

Imagen. 3.3. SELECT a la tabla PACIENTE.

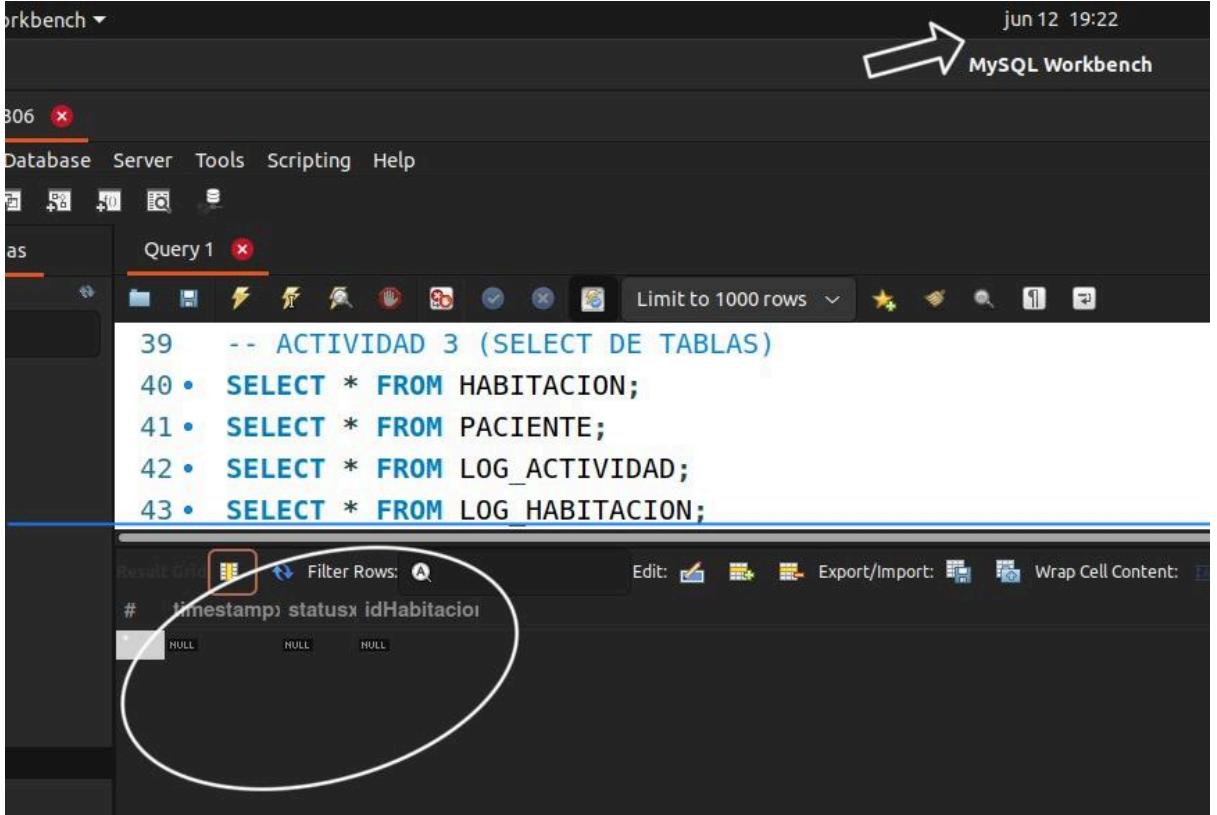
The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query editor contains the following code:

```
39 -- ACTIVIDAD 3 (SELECT DE TABLAS)
40 • SELECT * FROM HABITACION;
41 • SELECT * FROM PACIENTE;
42 • SELECT * FROM LOG_ACTIVIDAD;
43 • SELECT * FROM LOG_HABITACION;
```

The results grid displays data from the LOG_ACTIVIDAD table, with a large oval highlighting the first 18 rows. The columns are labeled: #, id_log_actividad, timestamppx, actividad, idPaciente, and idHabitacion.

#	id_log_actividad	timestamppx	actividad	idPaciente	idHabitacion
1	1	5/31/2021 7:12:14 AM	Paciente inicia el registro.	134247	10
2	2	5/31/2021 7:13:16 AM	Paciente recibe papeleria en recep...	134247	10
3	3	5/31/2021 7:15:54 AM	Paciente entrega papeleria.	134247	10
4	4	5/31/2021 7:16:33 AM	Repcionista establece la condicio...	134247	10
5	5	5/31/2021 7:17:47 AM	Enfermera comienza la revision del...	134247	10
6	6	5/31/2021 7:19:33 AM	Revision determino que el paciente ...	134247	10
7	7	5/31/2021 7:20:34 AM	Medico inicia con el tratamiento del ...	134247	10
8	8	5/31/2021 7:21:50 AM	Paciente inicia el registro.	135641	10
9	9	5/31/2021 7:23:01 AM	Paciente recibe papeleria en recep...	135641	10
10	10	5/31/2021 7:25:39 AM	Paciente inicia el registro.	180487	10
11	11	5/31/2021 7:25:44 AM	Paciente entrega papeleria.	135641	12
12	12	5/31/2021 7:26:18 AM	Repcionista establece la condici...	135641	12
13	13	5/31/2021 7:26:43 AM	Paciente recibe papeleria en recep...	180487	5
14	14	5/31/2021 7:27:33 AM	Enfermera comienza la revision del...	135641	5
15	15	5/31/2021 7:29:12 AM	Paciente entrega papeleria.	180487	7
16	16	5/31/2021 7:29:15 AM	Revision determino que el paciente ...	135641	7
17	17	5/31/2021 7:29:47 AM	Repcionista establece la condici...	180487	7
18	18	5/31/2021 7:30:34 AM	Medico inicia con el tratamiento del ...	135641	7

Imagen 3.4. SELECT a la tabla LOG_ACTIVIDAD.



The screenshot shows the MySQL Workbench interface. In the top right corner, it says "jun 12 19:22" and "MySQL Workbench". The title bar has "Workbench" and "Query 1". The menu bar includes "Database", "Server", "Tools", "Scripting", and "Help". Below the menu is a toolbar with various icons. The main area contains a query editor with the following code:

```

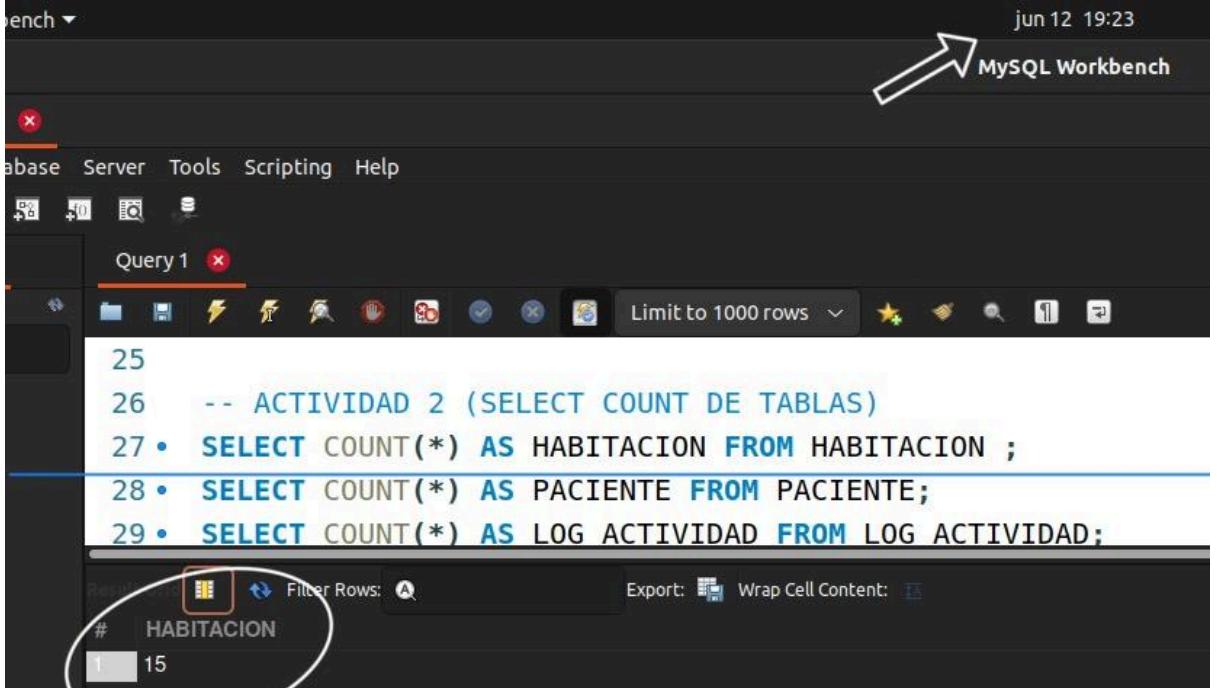
39 -- ACTIVIDAD 3 (SELECT DE TABLAS)
40 • SELECT * FROM HABITACION;
41 • SELECT * FROM PACIENTE;
42 • SELECT * FROM LOG_ACTIVIDAD;
43 • SELECT * FROM LOG_HABITACION;

```

Below the code is a results grid titled "Result Grid". It has columns: #, timestamp, statusx, idHabitacion. There is one row with values: #, NULL, NULL, NULL. A white oval highlights the first column of the results grid.

Imagen 3.5. SELECT a la tabla LOG_HABITACION.

Luego de haber realizado los SELECT podemos confirmar el número de registros en cada una de las tablas realizando un SELECT COUNT(*) a cada una de las tablas.



The screenshot shows the MySQL Workbench interface. In the top right corner, it says "jun 12 19:23" and "MySQL Workbench". The title bar has "Workbench" and "Query 1". The menu bar includes "Database", "Server", "Tools", "Scripting", and "Help". Below the menu is a toolbar with various icons. The main area contains a query editor with the following code:

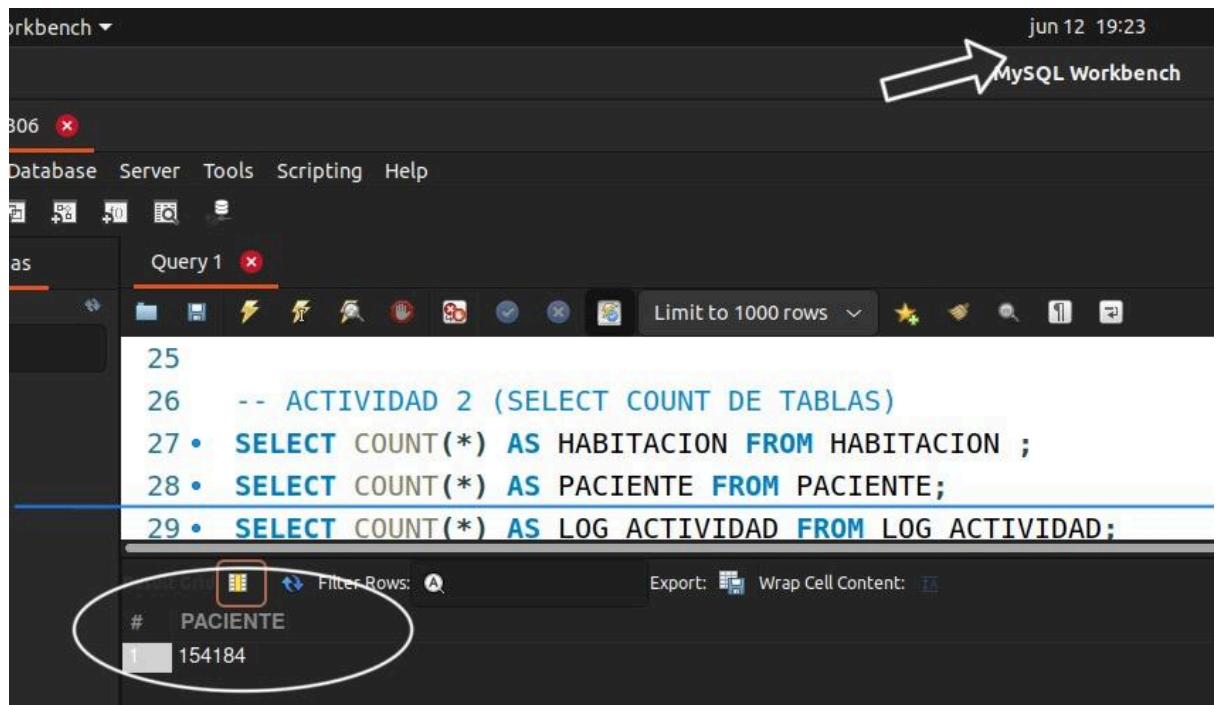
```

25
26 -- ACTIVIDAD 2 (SELECT COUNT DE TABLAS)
27 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
28 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE;
29 • SELECT COUNT(*) AS LOG ACTIVIDAD FROM LOG ACTIVIDAD;

```

Below the code is a results grid titled "Result Grid". It has columns: #, HABITACION. There is one row with values: #, 15. A white oval highlights the first column of the results grid.

Imagen 3.6. SELECT COUNT(*) a la tabla HABITACION.



MySQL Workbench - jun 12 19:23

306

Database Server Tools Scripting Help

Query 1

```

25
26 -- ACTIVIDAD 2 (SELECT COUNT DE TABLAS)
27 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
28 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE;
29 • SELECT COUNT(*) AS LOG ACTIVIDAD FROM LOG ACTIVIDAD;

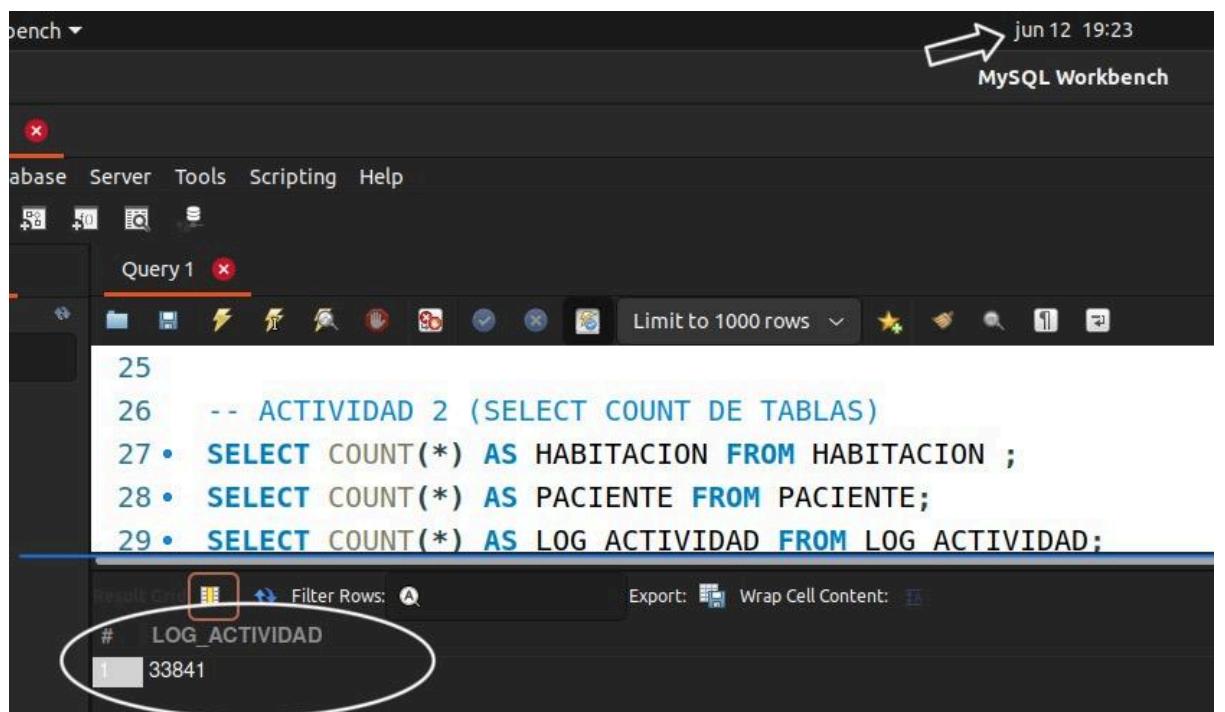
```

PACIENTE

#	PACIENTE
1	154184

Filter Rows: A Export: Wrap Cell Content: B

Imagen 3.7. SELECT COUNT(*) a la tabla PACIENTE.



MySQL Workbench - jun 12 19:23

306

Database Server Tools Scripting Help

Query 1

```

25
26 -- ACTIVIDAD 2 (SELECT COUNT DE TABLAS)
27 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
28 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE;
29 • SELECT COUNT(*) AS LOG ACTIVIDAD FROM LOG ACTIVIDAD;

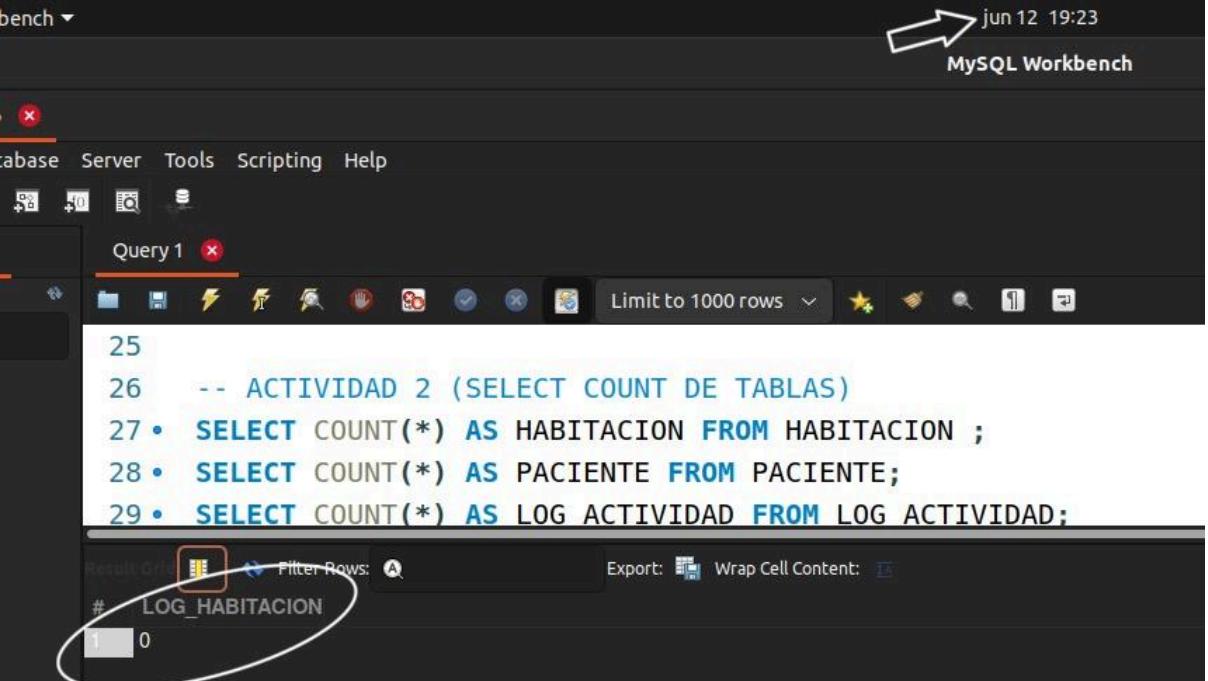
```

LOG_ACTIVIDAD

#	LOG_ACTIVIDAD
1	33841

Filter Rows: A Export: Wrap Cell Content: B

Imagen 3.8. SELECT COUNT(*) a la tabla LOG_ACTIVIDAD.



The screenshot shows the MySQL Workbench interface. In the top right corner, it says "jun 12 19:23" and "MySQL Workbench". The menu bar includes "Database", "Server", "Tools", "Scripting", and "Help". Below the menu is a toolbar with various icons. A query editor window titled "Query 1" contains the following SQL code:

```

25
26 -- ACTIVIDAD 2 (SELECT COUNT DE TABLAS)
27 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
28 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE;
29 • SELECT COUNT(*) AS LOG ACTIVIDAD FROM LOG ACTIVIDAD;

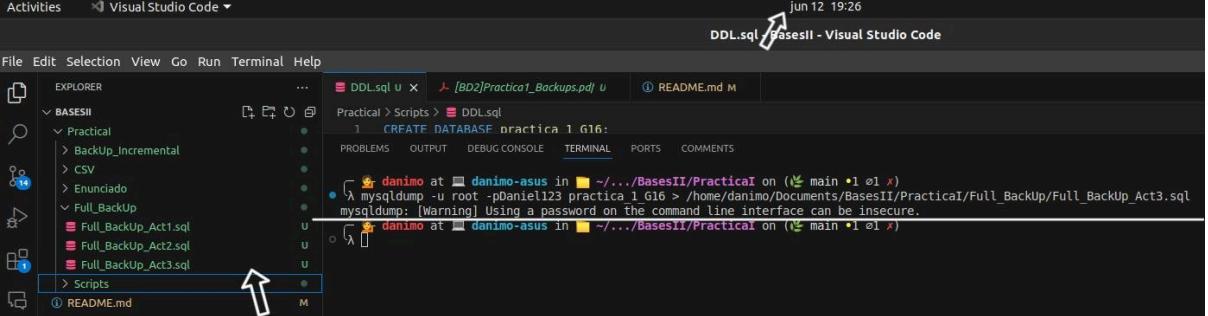
```

The results pane shows a table with one row:

#	LOG_HABITACION
1	0

Imagen 3.9. SELECT COUNT(*) a la tabla LOG_HABITACION.

Una vez que se verifica el número de registros en cada una de las tablas dentro de nuestra base de datos, el siguiente paso es la creación de backups. Primero creamos un full backup de nuestra base de datos.



The screenshot shows the Visual Studio Code interface. The title bar says "Activities > Visual Studio Code" and "jun 12 19:26". The main area shows an "EXPLORER" sidebar with a tree view of files and folders. In the center, there is a terminal window with the following text:

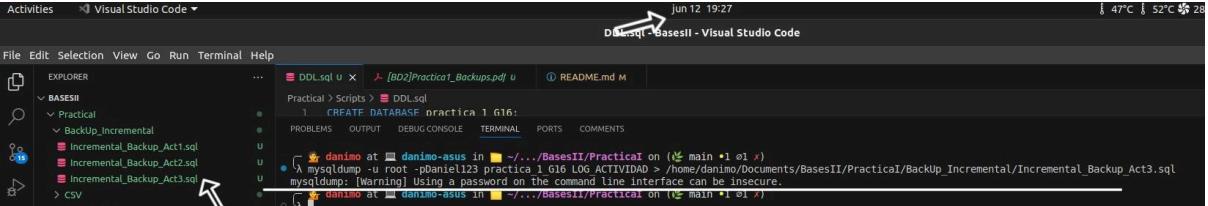
```

CREATE DATABASE practica_1 G16;
mysqldump: [Warning] Using a password on the command line interface can be insecure.

```

Imagen 3.10. Creación de Full backup de nuestra base de datos.

Luego de haber creado nuestro full backup podemos crear nuestro backup incremental sobre la tabla LOG_ACTIVIDAD.



The screenshot shows the Visual Studio Code interface. The title bar says "Activities > Visual Studio Code" and "jun 12 19:27". The main area shows an "EXPLORER" sidebar with a tree view of files and folders. In the center, there is a terminal window with the following text:

```

mysqldump: [Warning] Using a password on the command line interface can be insecure.

```

Imagen 3.11. Creación de Backup incremental de la tabla LOG_ACTIVIDAD.

Actividad 4

Al igual que la actividad anterior, en esta actividad cargaremos datos a la tabla LOG_ACTIVIDAD. Sin embargo, ahora lo haremos haciendo uso del archivo LogActividades2.

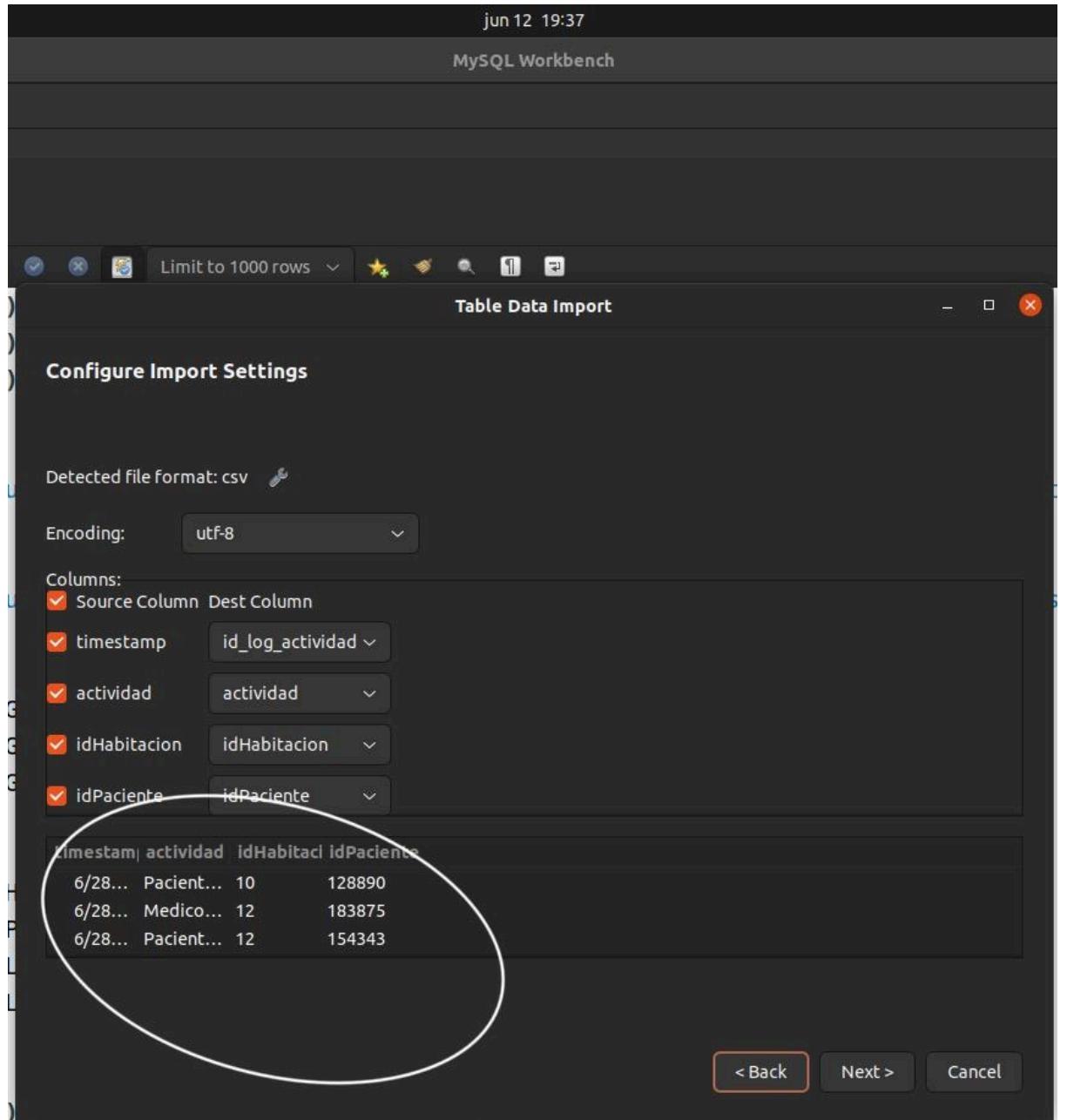


Imagen 4.1. Carga de datos a la tabla LOG_ACTIVIDAD.

Ahora procedemos a ejecutar las instrucciones SELECT a cada una de las tablas para así confirmar que la carga de datos se hizo correctamente.

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a menu bar with 'Base', 'Server', 'Tools', 'Scripting', and 'Help'. A status bar at the top right shows 'jun 12 19:59' and 'MySQL Workbench'. A white arrow points from the status bar towards the title bar. The main area has a tab labeled 'Query 1' which is currently selected. The query editor contains the following SQL code:

```
61
62 -- ACTIVIDAD 4 (SELECT DE TABLAS)
63 • SELECT * FROM HABITACION;
64 • SELECT * FROM PACIENTE;
65 • SELECT * FROM LOG_ACTIVIDAD;
66 • SELECT * FROM LOG_HABITACION;
67
```

Below the query editor is a result grid titled 'Result Grid'. It has a header row with columns '#', 'idHabitacion', and 'habitacion'. The data starts from row 1 and continues down to row 15. A large white circle highlights the first 15 rows of the result grid.

#	idHabitacion	habitacion
1	1	Sala de examenes 1
2	2	Sala de examenes 2
3	3	Sala de examenes 3
4	4	Sala de examenes 4
5	5	Sala de imagenes 1
6	6	Sala de procedimientos 1
7	7	Sala de procedimientos 2
8	8	Sala de procedimientos 3
9	9	Sala de procedimientos 4
10	10	Recepcion
11	11	Laboratorio
12	12	Estación de revisión 1
13	13	Estación de revisión 2
14	14	Estación de revisión 3
15	15	Estación de revisión 4
	NULL	NULL

Imagen 4.2. SELECT a la tabla HABITACION.

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a menu bar with 'Base', 'Server', 'Tools', 'Scripting', and 'Help'. A status bar at the top right indicates 'jun 12 19:59' and 'MySQL Workbench'. The main area is titled 'Query 1' and contains the following SQL code:

```
61
62 -- ACTIVIDAD 4 (SELECT DE TABLAS)
63 • SELECT * FROM HABITACION;
64 • SELECT * FROM PACIENTE;
65 • SELECT * FROM LOG_ACTIVIDAD;
66 • SELECT * FROM LOG_HABITACION;
67
```

Below the code is a 'Result Grid' section. It has a header row with columns '#', 'idPaciente', 'edad', and 'genero'. The data grid contains 15 rows of patient information. A large white circle highlights the first 15 rows of the result grid.

#	idPaciente	edad	genero
1	100000	95	Otro
2	100001	40	Femenino
3	100002	42	Masculino
4	100003	8	Femenino
5	100004	88	Masculino
6	100005	23	Masculino
7	100006	60	Femenino
8	100007	49	Femenino
9	100008	44	Femenino
10	100009	55	Femenino
11	100010	28	Femenino
12	100011	13	Femenino
13	100012	62	Masculino
14	100013	32	Masculino
15	100014	11	Masculino

Imagen 4.3. SELECT a la tabla PACIENTE.

The screenshot shows the MySQL Workbench interface. At the top right, there is a timestamp 'jun 12 19:59' and a red arrow pointing to the title 'MySQL Workbench'. The menu bar includes 'Database', 'Server', 'Tools', 'Scripting', and 'Help'. Below the menu is a toolbar with various icons. A tab labeled 'Query 1' is active, containing the following SQL code:

```

61
62      -- ACTIVIDAD 4 (SELECT DE TABLAS)
63 •  SELECT * FROM HABITACION;
64 •  SELECT * FROM PACIENTE;
65 •  SELECT * FROM LOG_ACTIVIDAD;
66 •  SELECT * FROM LOG_HABITACION;
67

```

The results are displayed in a 'Result Grid' table. The columns are labeled '#', 'id_log_activida', 'timestamp', 'actividad', 'idPaciente', and 'idHabitacion'. The data consists of 11 rows:

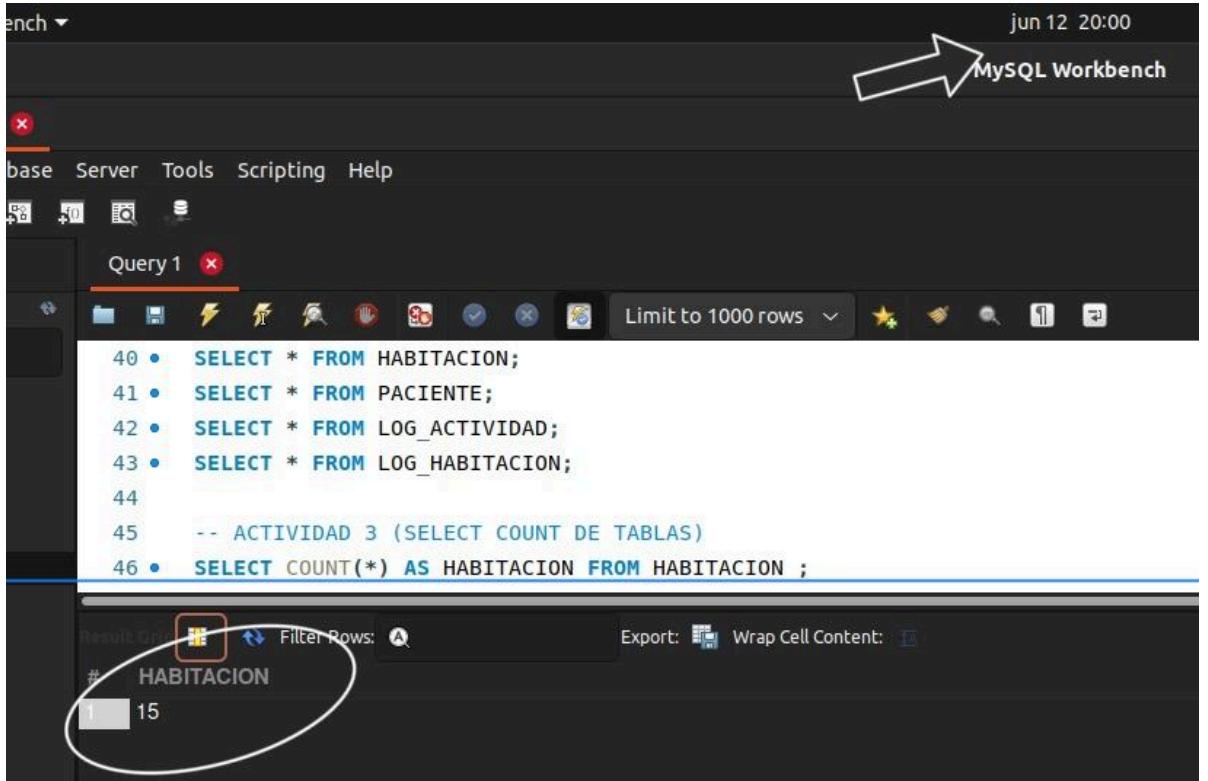
#	id_log_activida	timestamp	actividad	idPaciente	idHabitacion
1	1	5/31/2021 7:12:14 AM	Paciente inicia el registro.	134247	10
2	2	5/31/2021 7:13:16 AM	Paciente recibe papeleria en recep...	134247	10
3	3	5/31/2021 7:15:54 AM	Paciente entrega papeleria.	134247	10
4	4	5/31/2021 7:16:33 AM	Recepcionista establece la condicio...	134247	10
5	5	5/31/2021 7:17:47 AM	Enfermera comienza la revision del...	134247	10
6	6	5/31/2021 7:19:33 AM	Revision determino que el paciente ...	134247	10
7	7	5/31/2021 7:20:34 AM	Medico inicia con el tratamiento del ...	134247	10
8	8	5/31/2021 7:21:50 AM	Paciente inicia el registro.	135641	10
9	9	5/31/2021 7:23:01 AM	Paciente recibe papeleria en recep...	135641	10
10	10	5/31/2021 7:25:39 AM	Paciente inicia el registro.	180487	10
11	11	5/31/2021 7:25:44 AM	Paciente entrega papeleria.	135641	12

Imagen 4.4. SELECT a la tabla LOG_ACTIVIDAD.

This screenshot is similar to the previous one, showing the MySQL Workbench interface with the same timestamp and menu. The 'Query 1' tab contains the same SQL code. The results grid shows the same 11 rows of data. A white oval highlights the first row of the result set. The columns are labeled '#', 'timestamp', 'status', and 'idHabitacion'. The data for the first row is: # NULL, timestamp NULL, status NULL, idHabitacion NULL.

Imagen 4.5. SELECT a la tabla LOG_HABITACION.

Una vez confirmado esto, podemos confirmar el incremento de registros en nuestra tabla haciendo uso del comando SELECT COUNT(*) .



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

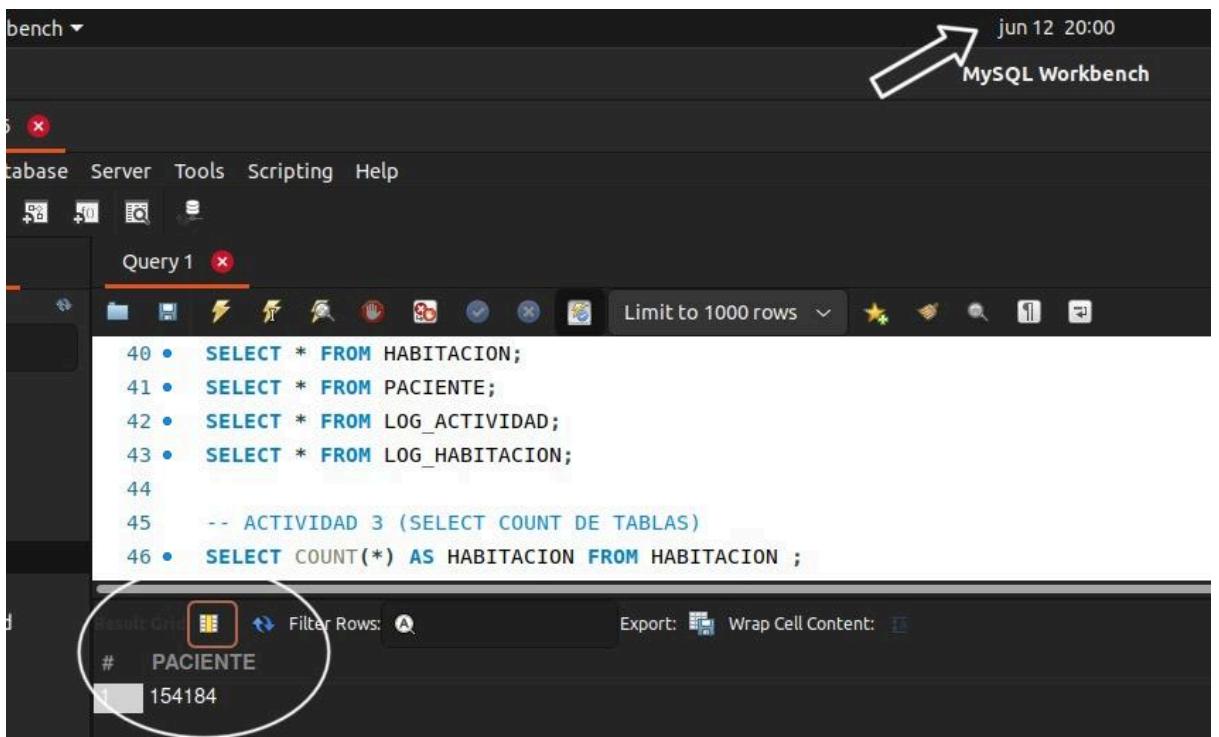
```

40 •   SELECT * FROM HABITACION;
41 •   SELECT * FROM PACIENTE;
42 •   SELECT * FROM LOG_ACTIVIDAD;
43 •   SELECT * FROM LOG_HABITACION;
44
45 -- ACTIVIDAD 3 (SELECT COUNT DE TABLAS)
46 •   SELECT COUNT(*) AS HABITACION FROM HABITACION ;

```

The results pane displays the output of the last query, which is a single row with the column '# HABITACION' and the value '15'. This row is circled with a white oval.

Imagen 4.6. SELECT COUNT(*) a la tabla HABITACION.



The screenshot shows the MySQL Workbench interface with a query editor containing the same SQL code as in Image 4.6:

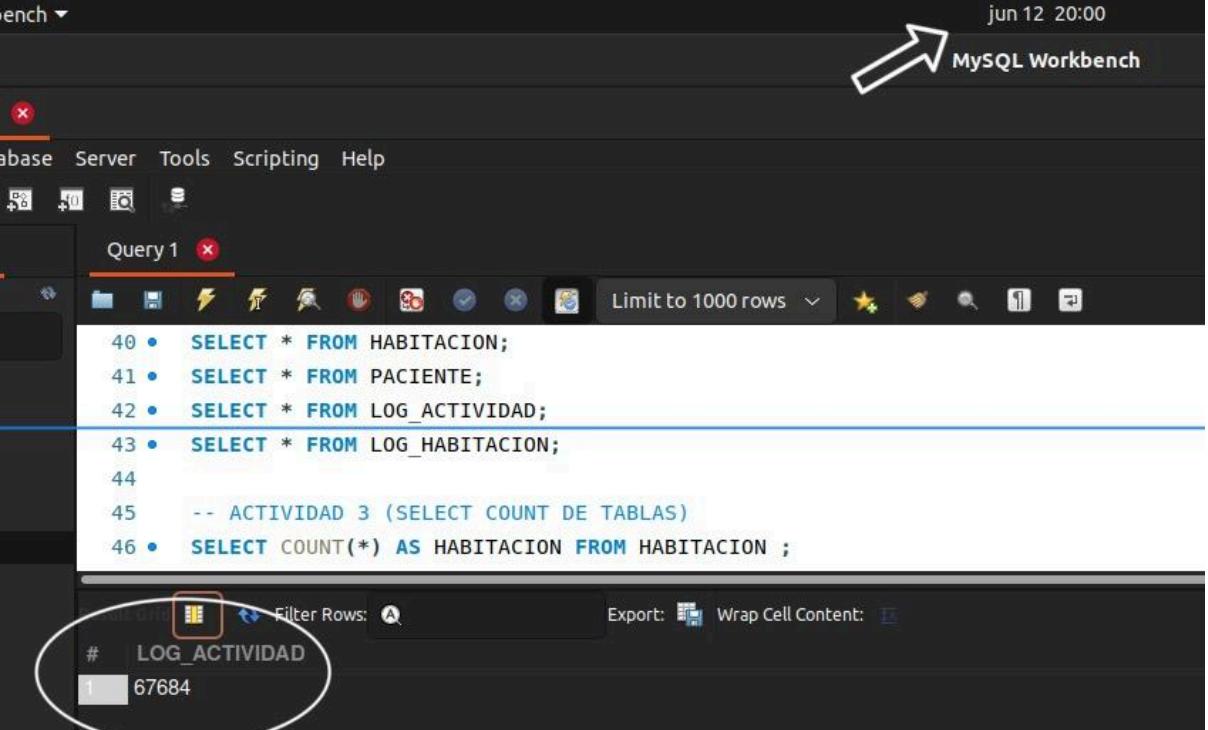
```

40 •   SELECT * FROM HABITACION;
41 •   SELECT * FROM PACIENTE;
42 •   SELECT * FROM LOG_ACTIVIDAD;
43 •   SELECT * FROM LOG_HABITACION;
44
45 -- ACTIVIDAD 3 (SELECT COUNT DE TABLAS)
46 •   SELECT COUNT(*) AS HABITACION FROM HABITACION ;

```

The results pane displays the output of the last query, which is a single row with the column '# PACIENTE' and the value '154184'. This row is circled with a white oval.

Imagen 4.7. SELECT COUNT(*) a la tabla PACIENTE.



The screenshot shows the MySQL Workbench interface with a query editor titled "Query 1". The code entered is:

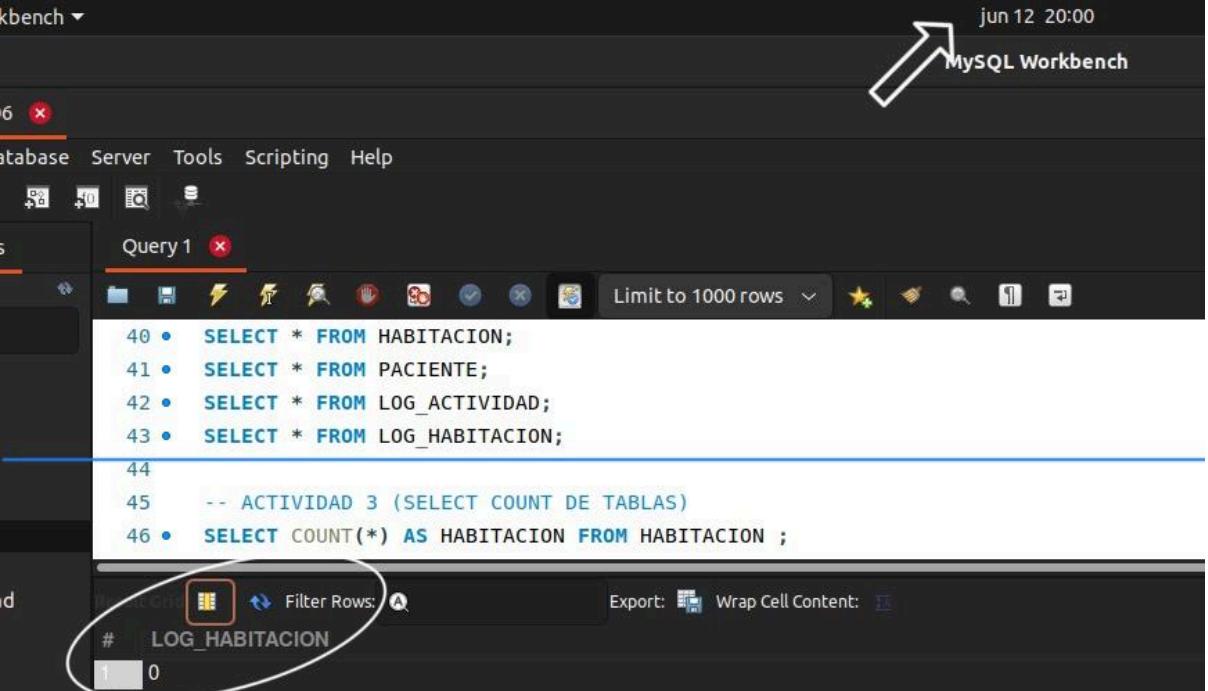
```

40 • SELECT * FROM HABITACION;
41 • SELECT * FROM PACIENTE;
42 • SELECT * FROM LOG_ACTIVIDAD;
43 • SELECT * FROM LOG_HABITACION;
44
45 -- ACTIVIDAD 3 (SELECT COUNT DE TABLAS)
46 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;

```

The results pane displays the output of the last query, which is a single row with one column. The column is labeled "# LOG_ACTIVIDAD" and contains the value "67684". This row is highlighted with a red oval.

Imagen 4.8. SELECT COUNT(*) a la tabla LOG_ACTIVIDAD.



The screenshot shows the MySQL Workbench interface with a query editor titled "Query 1". The code entered is identical to Image 4.7:

```

40 • SELECT * FROM HABITACION;
41 • SELECT * FROM PACIENTE;
42 • SELECT * FROM LOG_ACTIVIDAD;
43 • SELECT * FROM LOG_HABITACION;
44
45 -- ACTIVIDAD 3 (SELECT COUNT DE TABLAS)
46 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;

```

The results pane displays the output of the last query, which is a single row with one column. The column is labeled "# LOG_HABITACION" and contains the value "0". This row is highlighted with a red oval.

Imagen 4.9. SELECT COUNT(*) a la tabla LOG_HABITACION.

Ahora que hemos verificado que nuestras tablas posean los datos que acabamos de cargar. Podemos realizar nuestras copias de seguridad. Primero crearemos un full backup de nuestra base de datos para la actividad 4.

```
jun 12 20:01
DDL.sql - BasesII - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER BASESII Practical Scripts
    > BackUp_Incremental
    > CSV
    > Enunciado
    > Full_BackUp
        Full_BackUp_Act1.sql
        Full_BackUp_Act2.sql
        Full_BackUp_Act3.sql
        Full_BackUp_Act4.sql
    > Scripts
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
CREATE DATABASE practica_1 G16;
mysqldump: [Warning] Using a password on the command line interface can be insecure.
mysqldump: [Warning] Using a password on the command line interface can be insecure.
```

Imagen 4.10. Creación de full backup de nuestra base de datos.

Para finalizar la actividad 4 creamos el backup incremental sobre la tabla LOG_ACTIVIDAD.

```
jun 12 20:02
DDL.sql - BasesII - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER BASESII Practical Scripts
    > BackUp_Incremental
        Incremental_Backup_Act1.sql
        Incremental_Backup_Act2.sql
        Incremental_Backup_Act3.sql
        Incremental_Backup_Act4.sql
    > Scripts
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
CREATE DATABASE practica_1 G16;
mysqldump: [Warning] Using a password on the command line interface can be insecure.
mysqldump: [Warning] Using a password on the command line interface can be insecure.
```

Imagen 4.11. Creación de backup incremental de la tabla LOG_ACTIVIDAD.

Actividad 5

Para finalizar las actividades de creación de backups cargamos los datos en la tabla LOG_HABITACION.

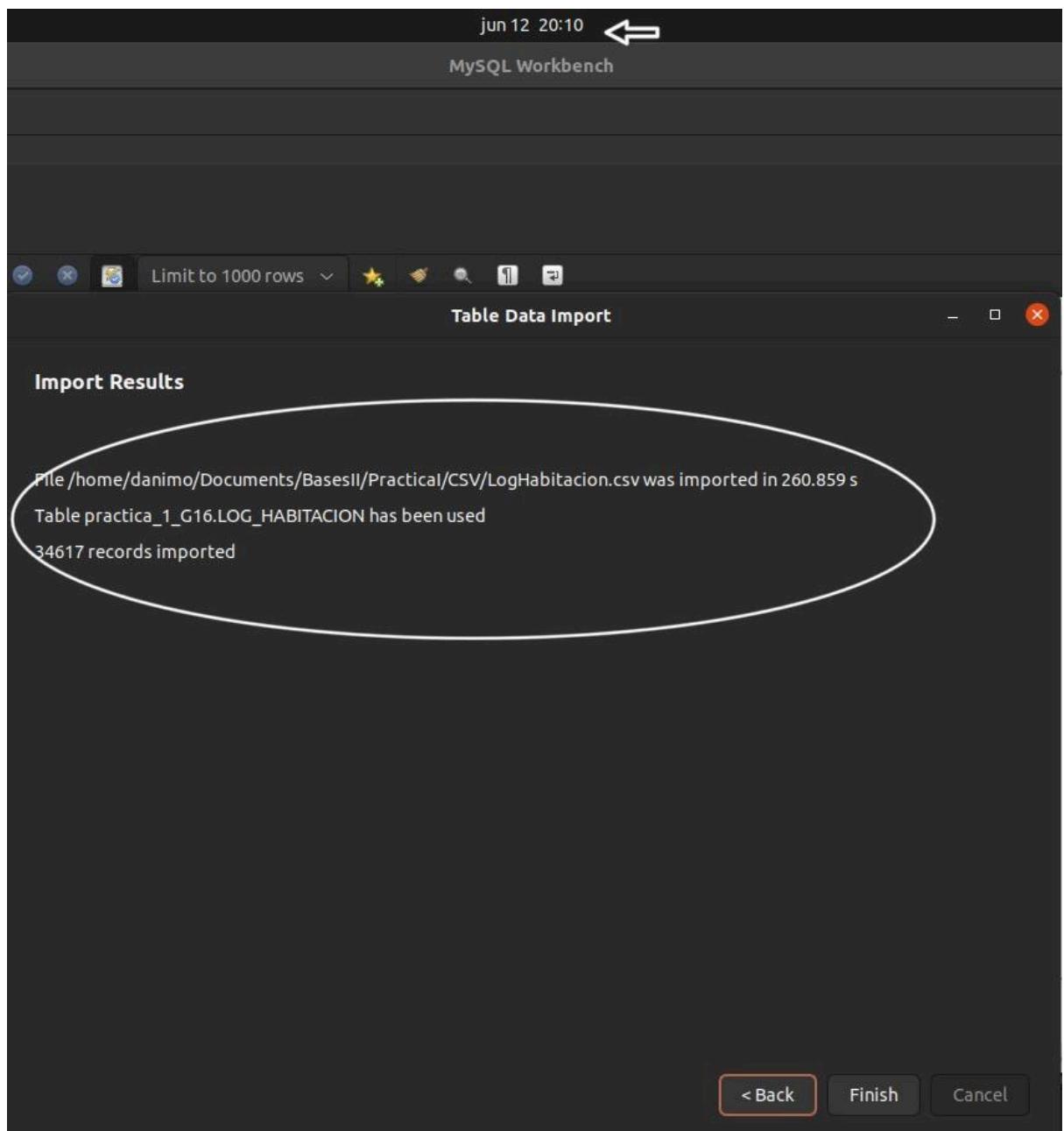
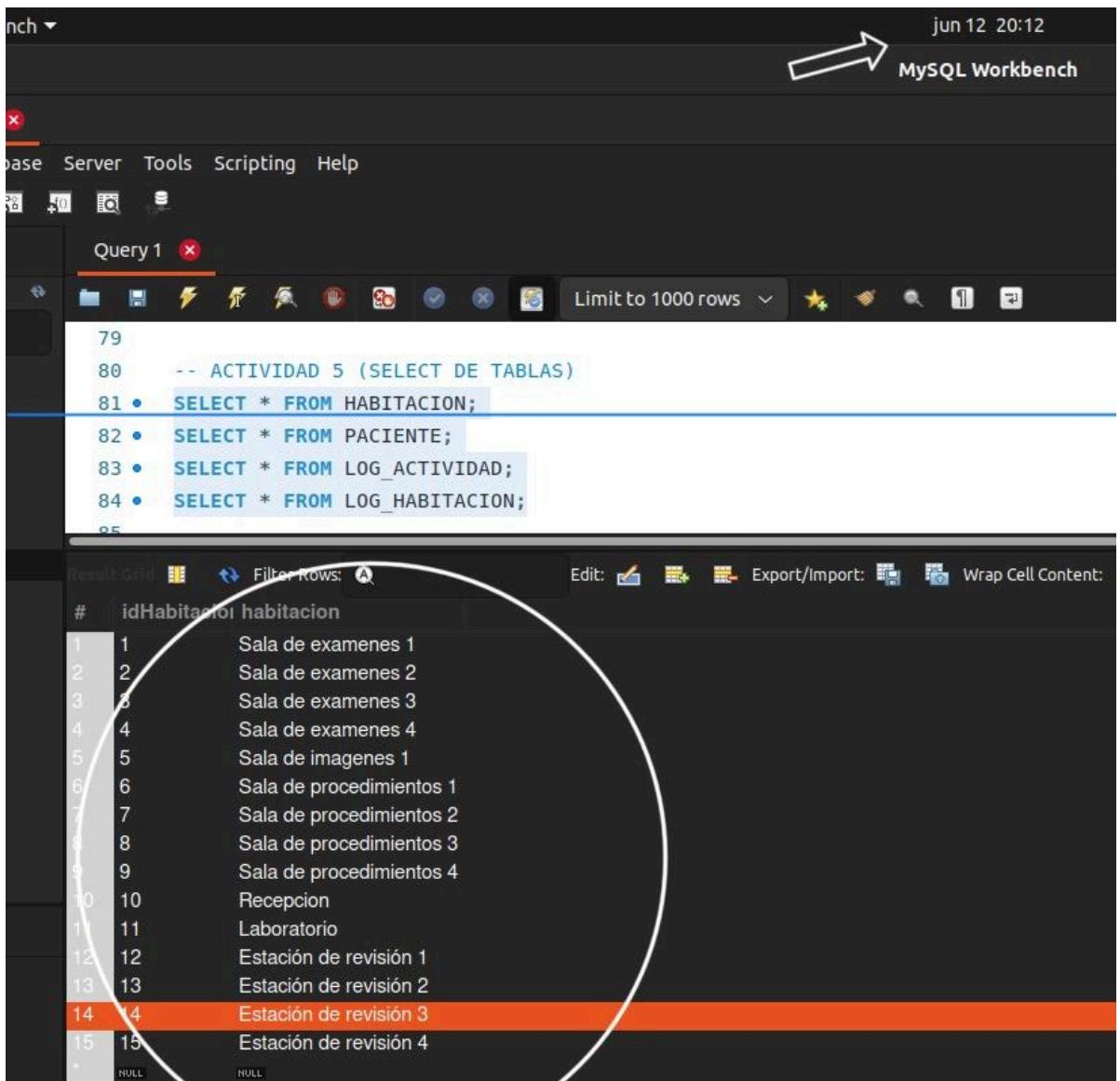


Imagen 5.1. Carga de datos a la tabla LOG_HABITACION.

Ahora procedemos a ejecutar las instrucciones SELECT a cada una de las tablas para así confirmar que la carga de datos se hizo correctamente.



The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a menu bar with 'Database', 'Server', 'Tools', 'Scripting', and 'Help'. A status bar at the top right shows the date 'jun 12 20:12' and the application name 'MySQL Workbench'. A red arrow points from the text below to the status bar.

Query 1

```
79
80  -- ACTIVIDAD 5 (SELECT DE TABLAS)
81 • SELECT * FROM HABITACION;
82 • SELECT * FROM PACIENTE;
83 • SELECT * FROM LOG_ACTIVIDAD;
84 • SELECT * FROM LOG_HABITACION;
85
```

Result Grid

#	idHabitacion	habitacion
1	1	Sala de examenes 1
2	2	Sala de examenes 2
3	3	Sala de examenes 3
4	4	Sala de examenes 4
5	5	Sala de imagenes 1
6	6	Sala de procedimientos 1
7	7	Sala de procedimientos 2
8	8	Sala de procedimientos 3
9	9	Sala de procedimientos 4
10	10	Recepcion
11	11	Laboratorio
12	12	Estación de revisión 1
13	13	Estación de revisión 2
14	14	Estación de revisión 3
15	15	Estación de revisión 4
*	NULL	NULL

A large white circle highlights the data in the 'habitacion' column. The row with idHabitacion 14 is highlighted with an orange background.

Imagen 5.2. SELECT a la tabla HABITACION.

The screenshot shows the MySQL Workbench interface. At the top right, there is a timestamp "jun 12 20:12" and the title "MySQL Workbench". A white arrow points from the timestamp towards the title. Below the title, there is a toolbar with various icons. The main area is titled "Query 1" and contains the following SQL code:

```
79
80    -- ACTIVIDAD 5 (SELECT DE TABLAS)
81 •  SELECT * FROM HABITACION;
82 •  SELECT * FROM PACIENTE;
83 •  SELECT * FROM LOG_ACTIVIDAD;
84 •  SELECT * FROM LOG_HABITACION;
85
```

Below the code, the "Result Grid" pane displays the results of the "SELECT * FROM PACIENTE;" query. The results are presented in a table with the following columns: #, idPaciente, edad, and genero. The data is as follows:

#	idPaciente	edad	genero
1	100000	95	Otro
2	100001	40	Femenino
3	100002	42	Masculino
4	100003	8	Femenino
5	100004	88	Masculino
6	100005	23	Masculino
7	100006	60	Femenino
8	100007	49	Femenino
9	100008	44	Femenino
10	100009	55	Femenino
11	100010	28	Femenino
12	100011	13	Femenino
13	100012	62	Masculino
14	100013	32	Masculino

Imagen 5.3. SELECT a la tabla PACIENTE.

Workbench ▾

jun 12 20:12

MySQL Workbench

3306 X

Database Server Tools Scripting Help

nas

Query 1 X

Limit to 1000 rows

79
80 -- ACTIVIDAD 5 (SELECT DE TABLAS)
81 • SELECT * FROM HABITACION;
82 • SELECT * FROM PACIENTE;
83 • SELECT * FROM LOG_ACTIVIDAD;
84 • SELECT * FROM LOG_HABITACION;

Result Grid Filter Rows: A Edit: Export/Import: Wrap Cell Content:

#	id_log_actividad	timestamp	actividad	idPaciente	idHabitacion
1	1	5/31/2021 7:12:14 AM	Paciente inicia el registro.	134247	10
2	2	5/31/2021 7:13:16 AM	Paciente recibe papeleria en recepcion...	134247	10
3	3	5/31/2021 7:15:54 AM	Paciente entrega papeleria.	134247	10
4	4	5/31/2021 7:16:33 AM	Recepcionista establece la condicion...	134247	10
5	5	5/31/2021 7:17:47 AM	Enfermera comienza la revision del...	134247	10
6	6	5/31/2021 7:19:33 AM	Revision determino que el paciente ...	134247	10
7	7	5/31/2021 7:20:34 AM	Medico inicia con el tratamiento del ...	134247	10
8	8	5/31/2021 7:21:50 AM	Paciente inicia el registro.	135641	10
9	9	5/31/2021 7:23:01 AM	Paciente recibe papeleria en recepcion...	135641	10
10	10	5/31/2021 7:25:39 AM	Paciente inicia el registro.	180487	10
11	11	5/31/2021 7:25:44 AM	Paciente entrega papeleria.	135641	12
12	12	5/31/2021 7:26:18 AM	Recepcionista establece la condicion...	135641	12
13	13	5/31/2021 7:26:43 AM	Paciente recibe papeleria en recepcion...	180487	5
14	14	5/31/2021 7:27:33 AM	Enfermera comienza la revision del...	135641	5
15	15	5/31/2021 7:29:12 AM	Paciente entrega papeleria.	180487	7

Imagen 5.4. SELECT a la tabla LOG_ACTIVIDAD.

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a menu bar with 'Database', 'Server', 'Tools', 'Scripting', and 'Help'. A tab labeled 'Query 1' is active. The main area contains a code editor with the following SQL script:

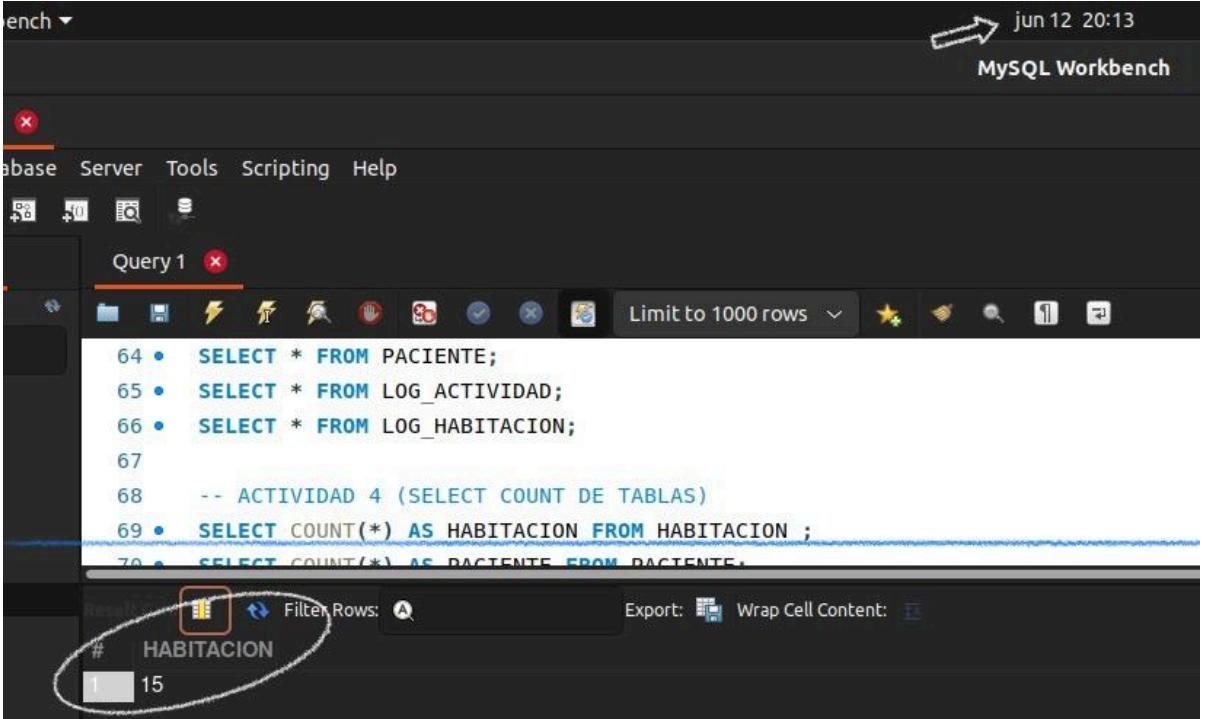
```
79
80    -- ACTIVIDAD 5 (SELECT DE TABLAS)
81 •  SELECT * FROM HABITACION;
82 •  SELECT * FROM PACIENTE;
83 •  SELECT * FROM LOG_ACTIVIDAD;
84 •  SELECT * FROM LOG_HABITACION;
```

Below the code editor is a results grid titled 'Result Grid'. It has columns for '#', 'timestamp', 'status', and 'idHabitacion'. The data is as follows:

#	timestamp	status	idHabitacion
1	5/31/2021 1:01:31 PM	Sala disponible.	6
2	5/31/2021 1:03:19 PM	Sala no disponible.	9
3	5/31/2021 1:04:19 PM	Inicia limpieza.	9
4	5/31/2021 1:08:20 PM	Sala disponible.	9
5	5/31/2021 1:10:15 PM	Sala no disponible.	13
6	5/31/2021 1:10:42 PM	Inicia limpieza.	13
7	5/31/2021 1:11:37 PM	Sala disponible.	13
8	5/31/2021 1:12:48 PM	Sala no disponible.	12
9	5/31/2021 1:13:14 PM	Inicia limpieza.	12
10	5/31/2021 1:15:14 PM	Sala disponible.	12
11	5/31/2021 1:17:11 PM	Sala no disponible.	14
12	5/31/2021 1:17:26 PM	Inicia limpieza.	14

Imagen 5.5. SELECT a la tabla LOG_HABITACION.

Una vez confirmado esto, podemos confirmar la cantidad de registros en nuestra tabla haciendo uso del comando `SELECT COUNT(*)`.



The screenshot shows the MySQL Workbench interface with a query editor titled "Query 1". The query executed is:

```

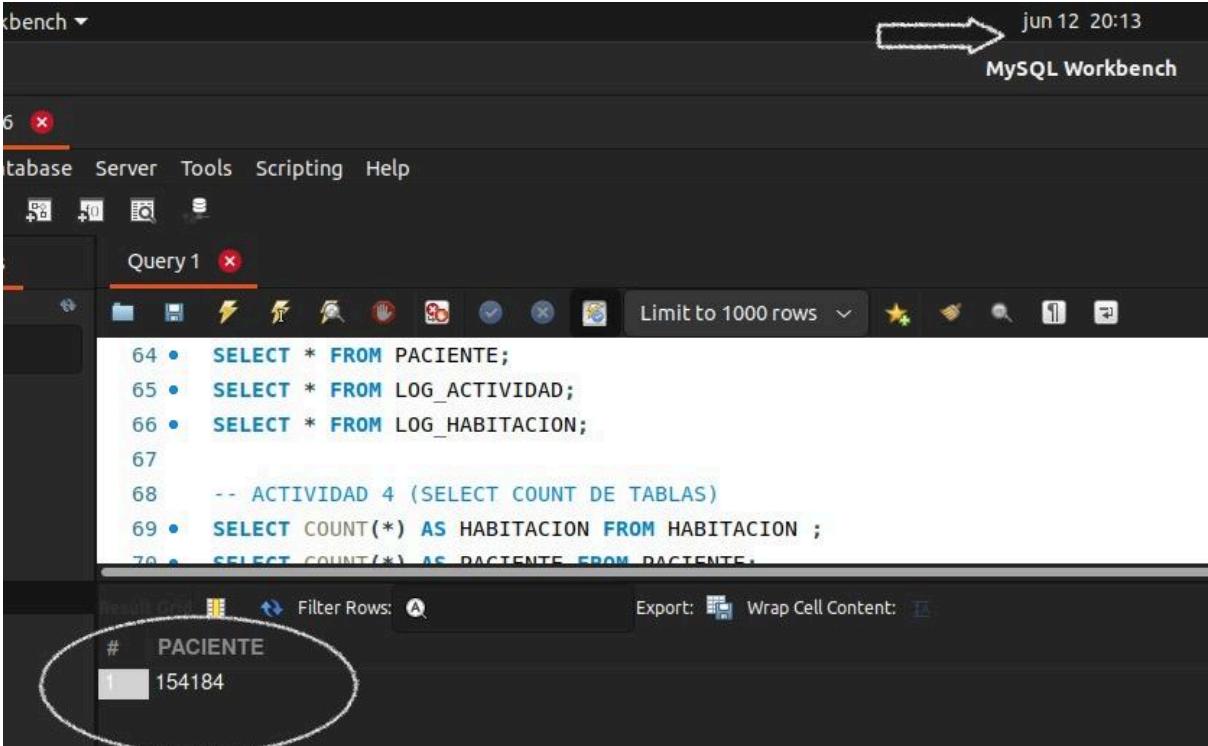
64 •  SELECT * FROM PACIENTE;
65 •  SELECT * FROM LOG_ACTIVIDAD;
66 •  SELECT * FROM LOG_HABITACION;
67
68 -- ACTIVIDAD 4 (SELECT COUNT DE TABLAS)
69 •  SELECT COUNT(*) AS HABITACION FROM HABITACION ;
70 •  SELECT COUNT(*) AS PACIENTE FROM PACIENTE;

```

The results pane displays the count for the HABITACION table:

#	HABITACION
1	15

Imagen 5.6. SELECT COUNT(*) a la tabla HABITACION.



The screenshot shows the MySQL Workbench interface with a query editor titled "Query 1". The query executed is:

```

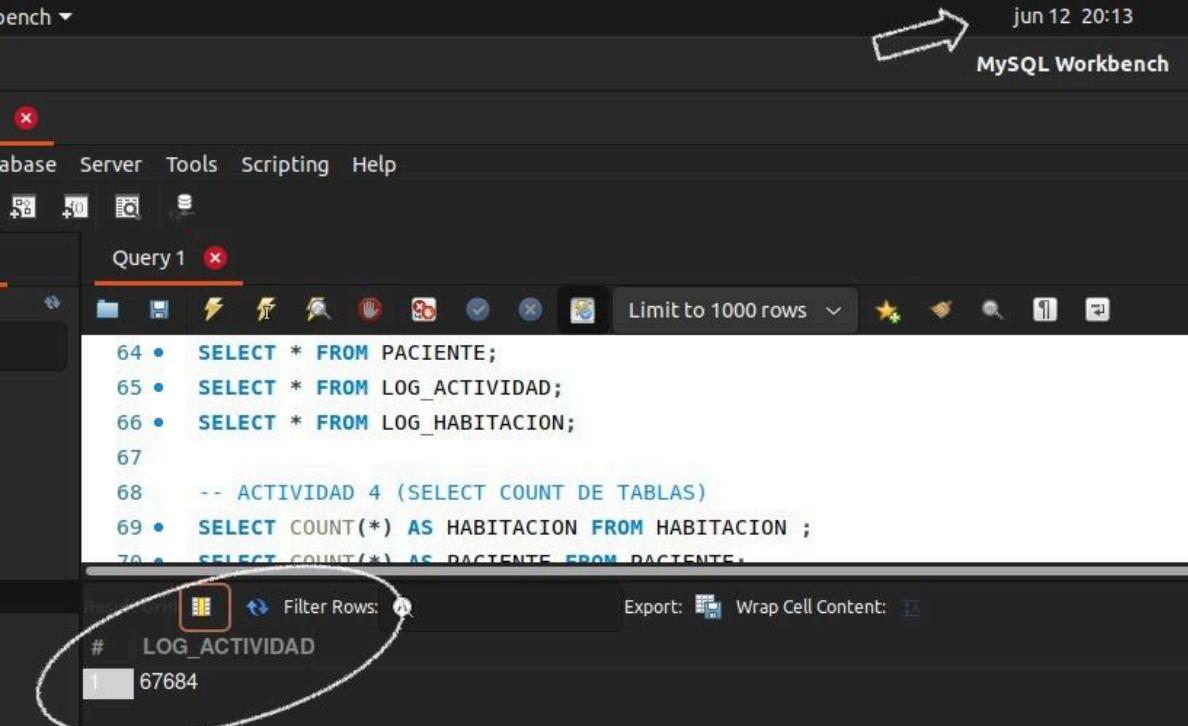
64 •  SELECT * FROM PACIENTE;
65 •  SELECT * FROM LOG_ACTIVIDAD;
66 •  SELECT * FROM LOG_HABITACION;
67
68 -- ACTIVIDAD 4 (SELECT COUNT DE TABLAS)
69 •  SELECT COUNT(*) AS HABITACION FROM HABITACION ;
70 •  SELECT COUNT(*) AS PACIENTE FROM PACIENTE;

```

The results pane displays the count for the PACIENTE table:

#	PACIENTE
1	154184

Imagen 5.7. SELECT COUNT(*) a la tabla PACIENTE.



The screenshot shows the MySQL Workbench interface with a query editor titled "Query 1". The query executed is:

```

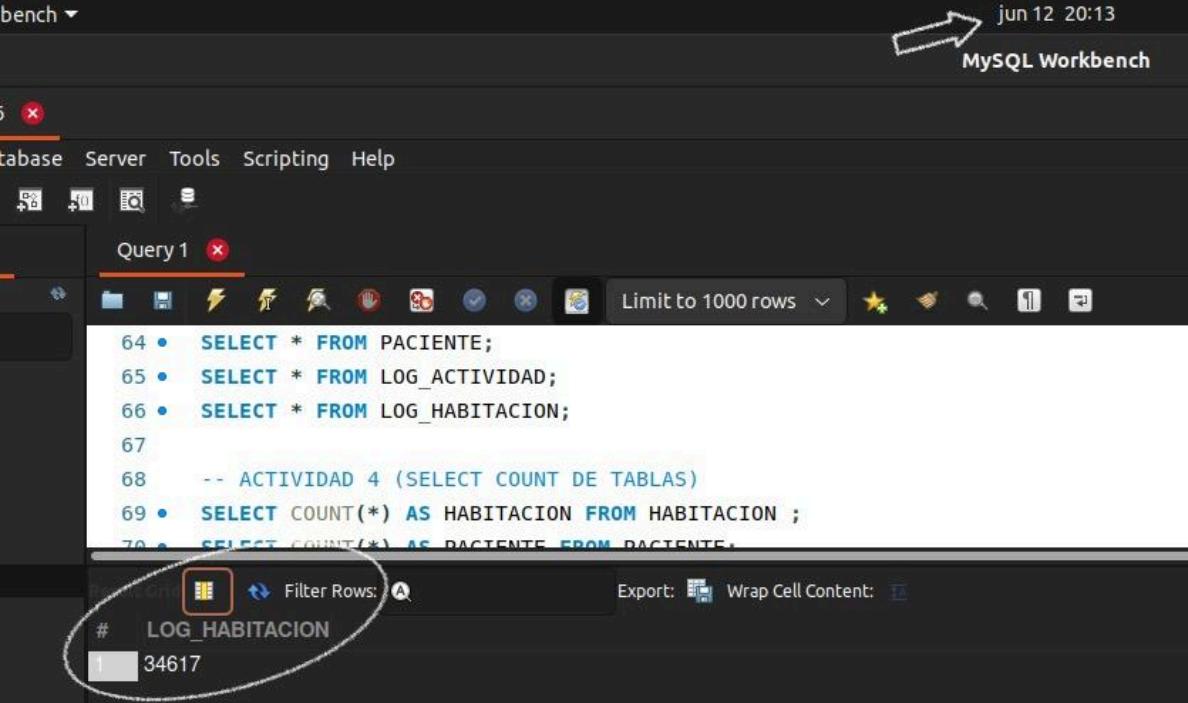
64 •  SELECT * FROM PACIENTE;
65 •  SELECT * FROM LOG_ACTIVIDAD;
66 •  SELECT * FROM LOG_HABITACION;
67
68 -- ACTIVIDAD 4 (SELECT COUNT DE TABLAS)
69 •  SELECT COUNT(*) AS HABITACION FROM HABITACION ;
70 •  SELECT COUNT(*) AS PACIENTE FROM PACIENTE;

```

The results pane displays the count for the LOG_ACTIVIDAD table:

#	LOG_ACTIVIDAD
1	67684

Imagen 5.8. SELECT COUNT(*) a la tabla LOG_ACTIVIDAD.



The screenshot shows the MySQL Workbench interface with a query editor titled "Query 1". The query executed is:

```

64 •  SELECT * FROM PACIENTE;
65 •  SELECT * FROM LOG_ACTIVIDAD;
66 •  SELECT * FROM LOG_HABITACION;
67
68 -- ACTIVIDAD 4 (SELECT COUNT DE TABLAS)
69 •  SELECT COUNT(*) AS HABITACION FROM HABITACION ;
70 •  SELECT COUNT(*) AS PACIENTE FROM PACIENTE;

```

The results pane displays the count for the LOG_HABITACION table:

#	LOG_HABITACION
1	34617

Imagen 5.9. SELECT COUNT(*) a la tabla LOG_HABITACION.

Ahora que hemos verificado que nuestras tablas posean los datos que acabamos de cargar. Podemos realizar nuestras copias de seguridad. Primero crearemos un full backup de nuestra base de datos para la actividad 5.

```
[danimo@danimo-asus ~] $ mysqldump -u root -pDanel123 practica1 G16 > /home/danimo/Documents/BasesII/PracticaI/Full_BackUp/Full_BackUp_Act5.sql
mysqldump: [Warning] Using a password on the command line interface can be insecure.
```

Imagen 5.10. Creación de full backup de nuestra base de datos.

Para finalizar la actividad 5 creamos el backup incremental sobre la tabla LOG_HABITACION.

```
[danimo@danimo-asus ~] $ mysqldump -u root -pDanel123 practica1 LOG_HABITACION > /home/danimo/Documents/BasesII/PracticaI/Backup_Incremental/Incremental_Backup_Act5.sql
mysqldump: [Warning] Using a password on the command line interface can be insecure.
```

Imagen 5.11. Creación de backup incremental de la tabla LOG_HABITACION.

Restauración de Full backup (Actividades 6-10)

Actividad 6

Para la primera actividad correspondiente a la restauración de full backups comenzamos eliminando los datos de nuestra base de datos. En este primer caso decidimos probar eliminando directamente la base de datos.

The screenshot shows a Visual Studio Code interface. At the top, there's a status bar with the date "jun 13 10:17" and the title "[BD2]Practica1_Backups.pdf - BasesII - Visual Studio Code". Below the status bar is a navigation bar with tabs for "Practical > Enunciado" and "[BD2]Practica1_Backups.pdf". The main area contains a PDF viewer showing page 5 of 7. To the right of the PDF viewer is a list of tasks:

- Carga de datos
- SELECT * FROM cada tabla
- SELECT COUNT(*) FROM cada tabla
- Creación de backup completo
- Creación de backup incremental

Restauración de Full backup

- Actividad 6:
 - Eliminación de datos
 - Restauración de full backup 1
 - SELECT * FROM cada tabla
 - SELECT COUNT(*) FROM cada tabla
- Actividad 7:
 - Eliminación de datos
 - Restauración de full backup 2
 - SELECT * FROM cada tabla
 - SELECT COUNT(*) FROM cada tabla
- Actividad 8:

In the bottom right corner of the terminal window, there is a large hand-drawn circle.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| actividad_one |
| educacion |
| information_schema |
| mysql |
| performance_schema |
| sys |
| tiempo_maya |
| tiempomaya |
| turismo_gt |
| universidad |
+-----+
10 rows in set (0.00 sec)
  
```

Imagen 6.1. Eliminación de la base de datos.

Luego de eliminar la base de datos de MySQL. Procedemos a crearla nuevamente haciendo uso del comando CREATE DATABASE. Después de la creación de la base de datos de nuevo, procedemos a ejecutar el comando para restaurar nuestra base de datos haciendo uso del full backup creado en la Actividad 1.

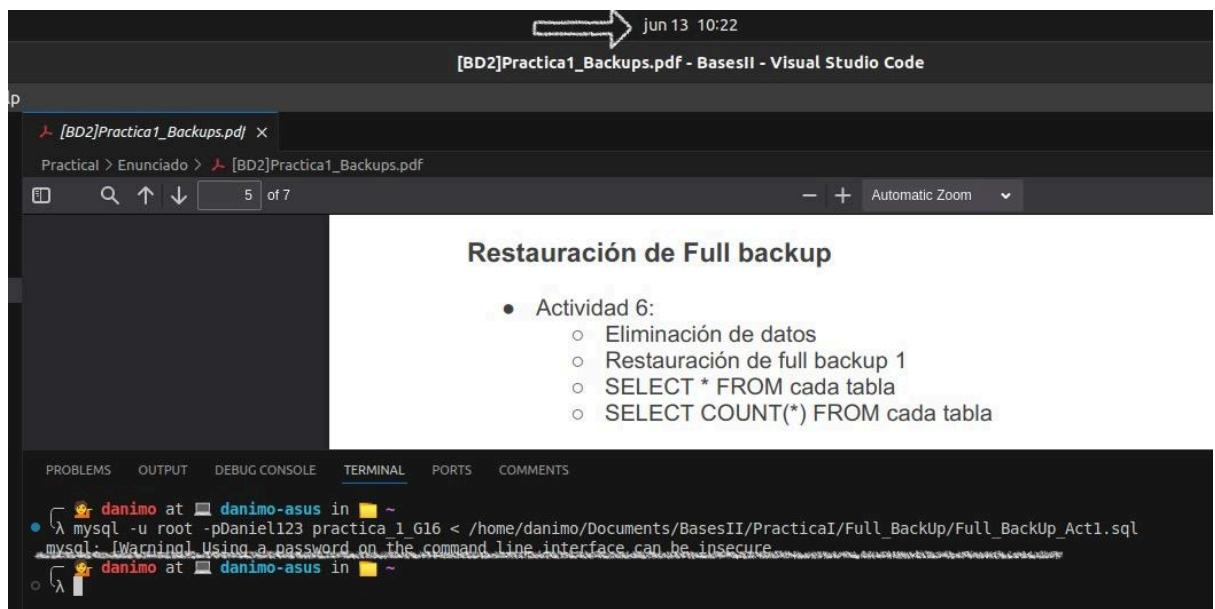


Imagen 6.2. Ejecución de comando para restaurar nuestro full backup 1.

Para verificar que se han cargado los datos de manera correcta, ahora ejecutamos el comando SELECT a cada una de nuestras tablas.

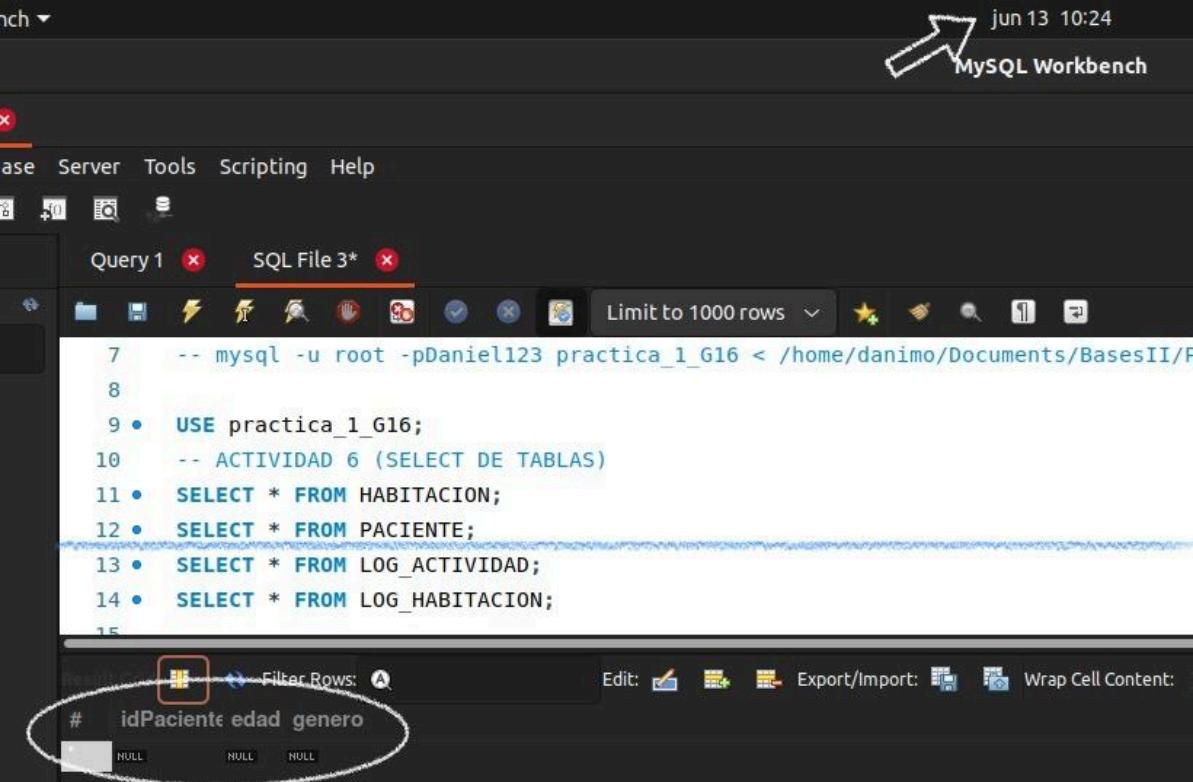
The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. The title bar says "jun 13 10:24 MySQL Workbench". Below the toolbar, the menu bar includes "File", "Server", "Tools", "Scripting", and "Help". There are two tabs open: "Query 1" and "SQL File 3*". The SQL tab contains the following code:

```
7 -- mysql -u root -pDaniel123 practica_1_G16 < /home/danimo/Documents/BasesII/Practica
8
9 • USE practica_1_G16;
10 -- ACTIVIDAD 6 (SELECT DE TABLAS)
11 • SELECT * FROM HABITACION;
12 • SELECT * FROM PACIENTE;
13 • SELECT * FROM LOG_ACTIVIDAD;
14 • SELECT * FROM LOG_HABITACION;
15
```

Below the code, the "Result Grid" pane displays the results of the "SELECT * FROM HABITACION;" query. The results are as follows:

#	idHabitacion	habitacion
1	1	Sala de examenes 1
2	2	Sala de examenes 2
3	3	Sala de examenes 3
4	4	Sala de examenes 4
5	5	Sala de imagenes 1
6	6	Sala de procedimientos 1
7	7	Sala de procedimientos 2
8	8	Sala de procedimientos 3
9	9	Sala de procedimientos 4
10	10	Repcion
11	11	Laboratorio
12	12	Estación de revisión 1
13	13	Estación de revisión 2
14	14	Estación de revisión 3
15	15	Estación de revisión 4
	NULL	NULL

Imagen 6.3. SELECT a la tabla HABITACION.



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

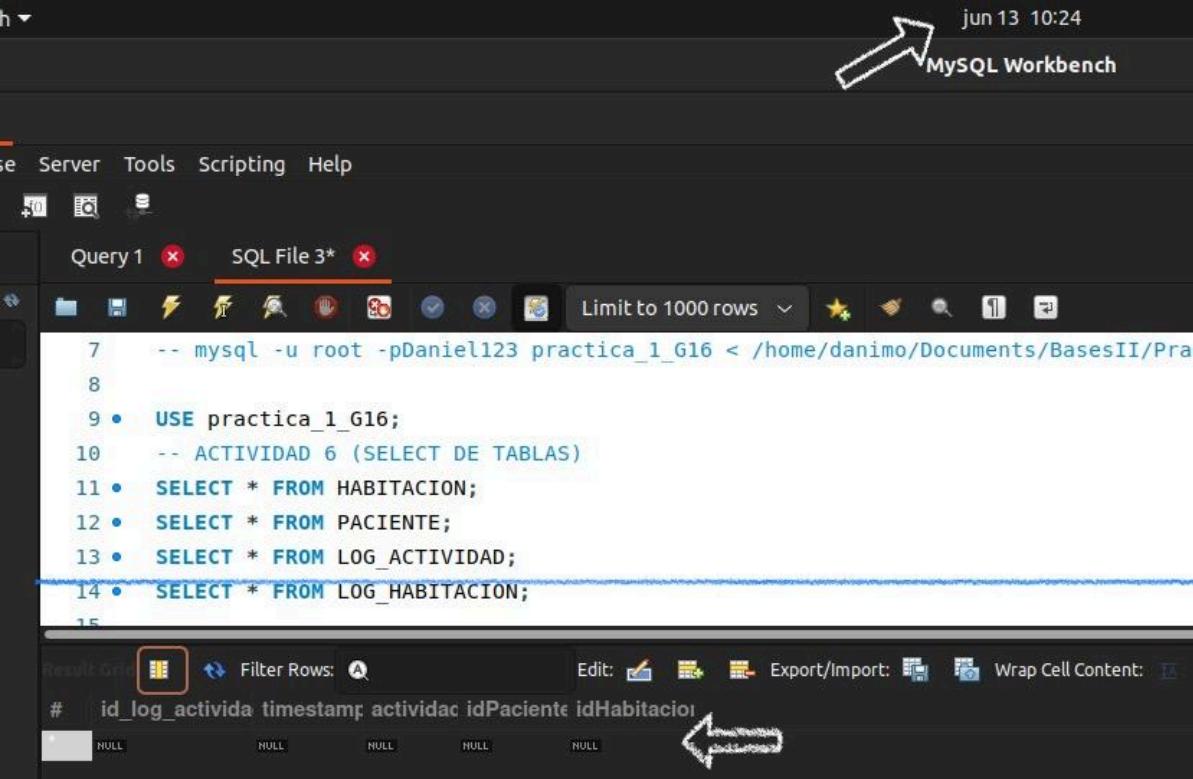
7 -- mysql -u root -pDaniel123 practica_1_G16 < /home/danimo/Documents/BasesII/Practica_1_G16.sql
8
9 • USE practica_1_G16;
10 -- ACTIVIDAD 6 (SELECT DE TABLAS)
11 • SELECT * FROM HABITACION;
12 • SELECT * FROM PACIENTE;
13 • SELECT * FROM LOG_ACTIVIDAD;
14 • SELECT * FROM LOG_HABITACION;
15

```

The results pane displays the following data for the PACIENTE table:

#	idPaciente	edad	genero
	NULL	NULL	NULL

Imagen 6.4. SELECT a la tabla PACIENTE.



The screenshot shows the MySQL Workbench interface with a query editor containing the same SQL code as in Image 6.4:

```

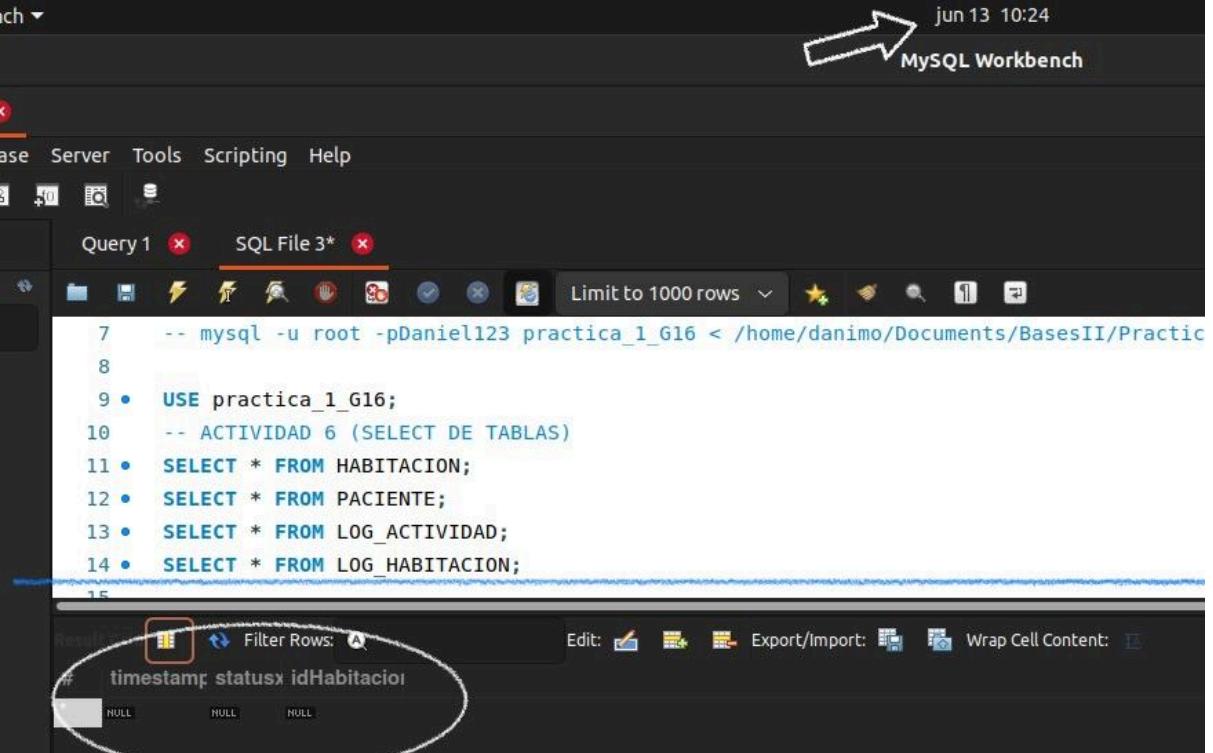
7 -- mysql -u root -pDaniel123 practica_1_G16 < /home/danimo/Documents/BasesII/Practica_1_G16.sql
8
9 • USE practica_1_G16;
10 -- ACTIVIDAD 6 (SELECT DE TABLAS)
11 • SELECT * FROM HABITACION;
12 • SELECT * FROM PACIENTE;
13 • SELECT * FROM LOG_ACTIVIDAD;
14 • SELECT * FROM LOG_HABITACION;
15

```

The results pane displays the following data for the LOG_ACTIVIDAD table:

#	id_log_actividad	timestamp	actividad	idPaciente	idHabitacion
	NULL	NULL	NULL	NULL	NULL

Imagen 6.5. SELECT a la tabla LOG_ACTIVIDAD.



The screenshot shows the MySQL Workbench interface. In the top right corner, there is a watermark-style arrow pointing right with the text "jun 13 10:24" and "MySQL Workbench". The main window has a dark theme. At the top, there is a menu bar with "Base", "Server", "Tools", "Scripting", and "Help". Below the menu is a toolbar with various icons. Two tabs are open: "Query 1" and "SQL File 3*". The "Query 1" tab contains the following SQL code:

```

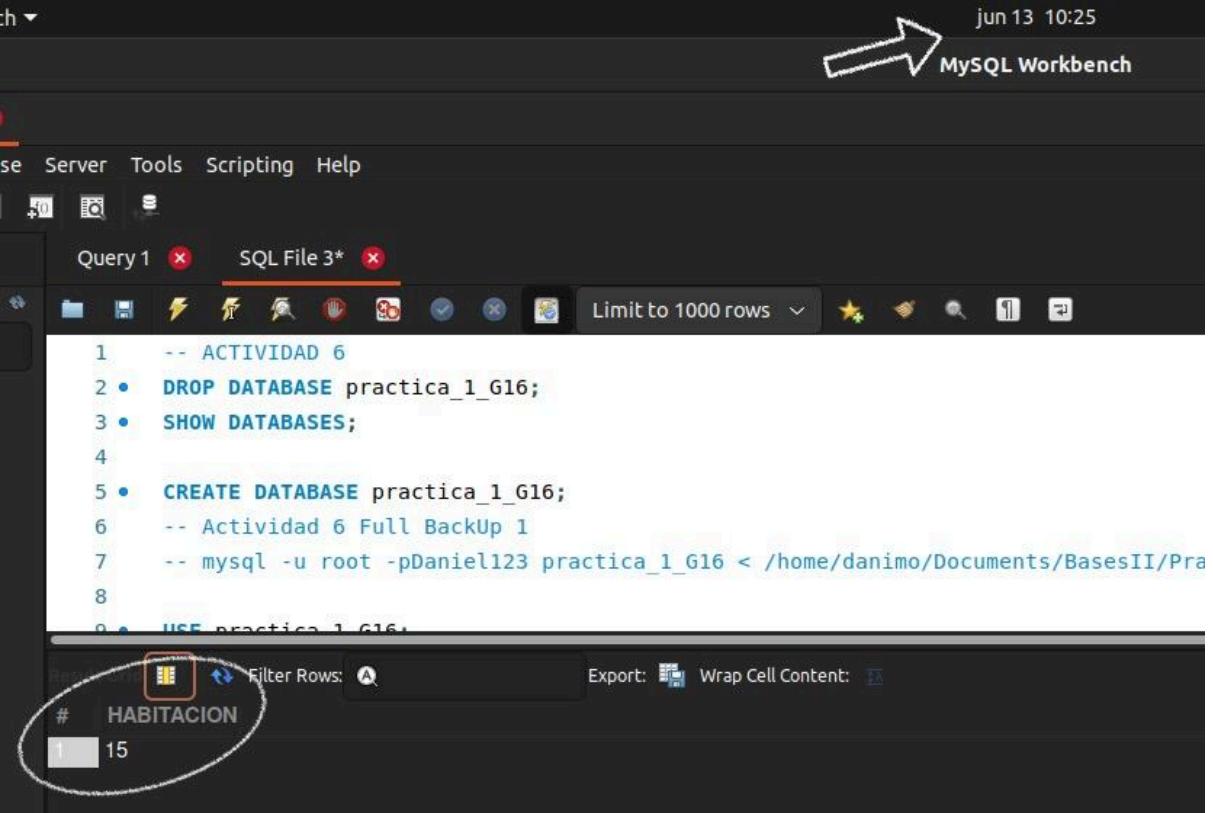
7 -- mysql -u root -pDaniel123 practica_1_G16 < /home/danim0/Documents/BasesII/Practic
8
9 • USE practica_1_G16;
10 -- ACTIVIDAD 6 (SELECT DE TABLAS)
11 • SELECT * FROM HABITACION;
12 • SELECT * FROM PACIENTE;
13 • SELECT * FROM LOG_ACTIVIDAD;
14 • SELECT * FROM LOG_HABITACION;
15

```

Below the code, the results pane shows a table with three columns: "#", "timestamp", and "statusx". The first row has values: # is 1, timestamp is NULL, and statusx is NULL. A red oval highlights the column headers and the first row of data.

Imagen 6.6. SELECT a la tabla LOG_HABITACION.

Además de esto, como medida de seguridad ejecutamos el comando SELECT COUNT(*) sobre cada una de nuestras tablas.



The screenshot shows the MySQL Workbench interface. In the top right corner, there is a watermark-style arrow pointing right with the text "jun 13 10:25" and "MySQL Workbench". The main window has a dark theme. At the top, there is a menu bar with "Base", "Server", "Tools", "Scripting", and "Help". Below the menu is a toolbar with various icons. Two tabs are open: "Query 1" and "SQL File 3*". The "Query 1" tab contains the following SQL code:

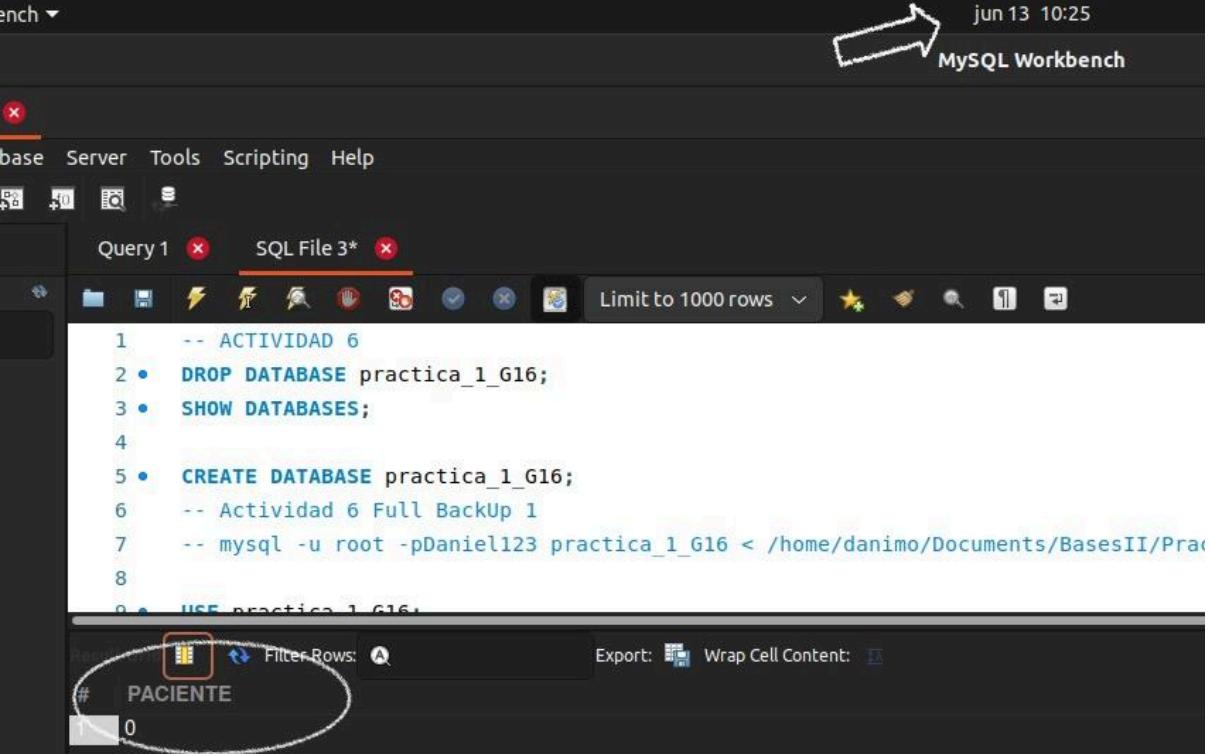
```

1 -- ACTIVIDAD 6
2 • DROP DATABASE practica_1_G16;
3 • SHOW DATABASES;
4
5 • CREATE DATABASE practica_1_G16;
6 -- Actividad 6 Full BackUp 1
7 -- mysql -u root -pDaniel123 practica_1_G16 < /home/danim0/Documents/BasesII/Prac
8
9 • USE practica_1_G16;

```

Below the code, the results pane shows a table with two columns: "#" and "HABITACION". The first row has values: # is 1 and HABITACION is 15. A red oval highlights the column headers and the first row of data.

Imagen 6.7. SELECT COUNT(*) a la tabla HABITACION.



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

1 -- ACTIVIDAD 6
2 • DROP DATABASE practica_1_G16;
3 • SHOW DATABASES;
4
5 • CREATE DATABASE practica_1_G16;
6 -- Actividad 6 Full BackUp 1
7 -- mysql -u root -pDaniel123 practica_1_G16 < /home/danim0/Documents/BasesII/Prac
8
9 • USE practica_1_G16;

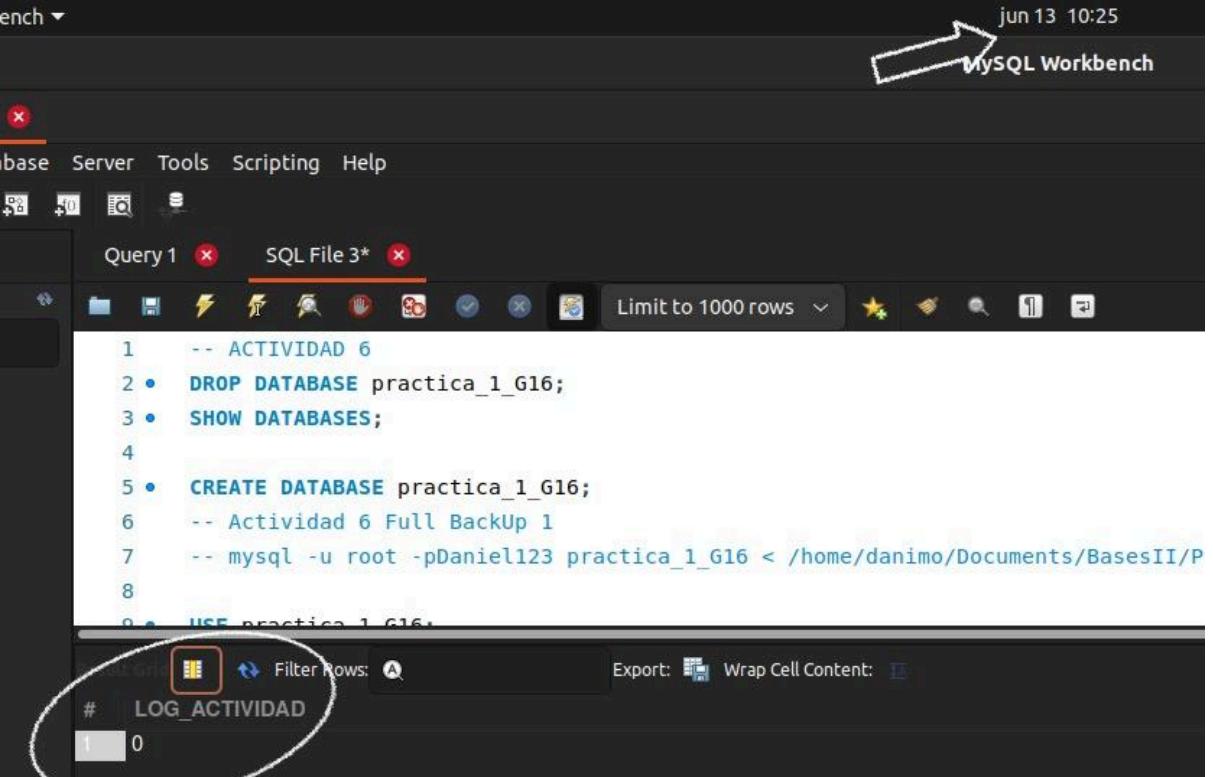
```

Below the code, the results of a query are displayed in a table:

#	PACIENTE
1	0

A white oval highlights the column header "PACIENTE". A white arrow points from the text "Imagen 6.8. SELECT COUNT(*) a la tabla PACIENTE." to the "PACIENTE" column header.

Imagen 6.8. SELECT COUNT(*) a la tabla PACIENTE.



The screenshot shows the MySQL Workbench interface with a query editor containing the same SQL code as in Image 6.7:

```

1 -- ACTIVIDAD 6
2 • DROP DATABASE practica_1_G16;
3 • SHOW DATABASES;
4
5 • CREATE DATABASE practica_1_G16;
6 -- Actividad 6 Full BackUp 1
7 -- mysql -u root -pDaniel123 practica_1_G16 < /home/danim0/Documents/BasesII/Prac
8
9 • USE practica_1_G16;

```

Below the code, the results of a query are displayed in a table:

#	LOG_ACTIVIDAD
1	0

A white oval highlights the column header "LOG_ACTIVIDAD". A white arrow points from the text "Imagen 6.9. SELECT COUNT(*) a la tabla LOG_ACTIVIDAD." to the "LOG_ACTIVIDAD" column header.

Imagen 6.9. SELECT COUNT(*) a la tabla LOG_ACTIVIDAD.

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it, a menu bar with 'Database', 'Server', 'Tools', 'Scripting', and 'Help'. There are two tabs open: 'Query 1' and 'SQL File 3*'. The 'Query 1' tab contains the following SQL code:

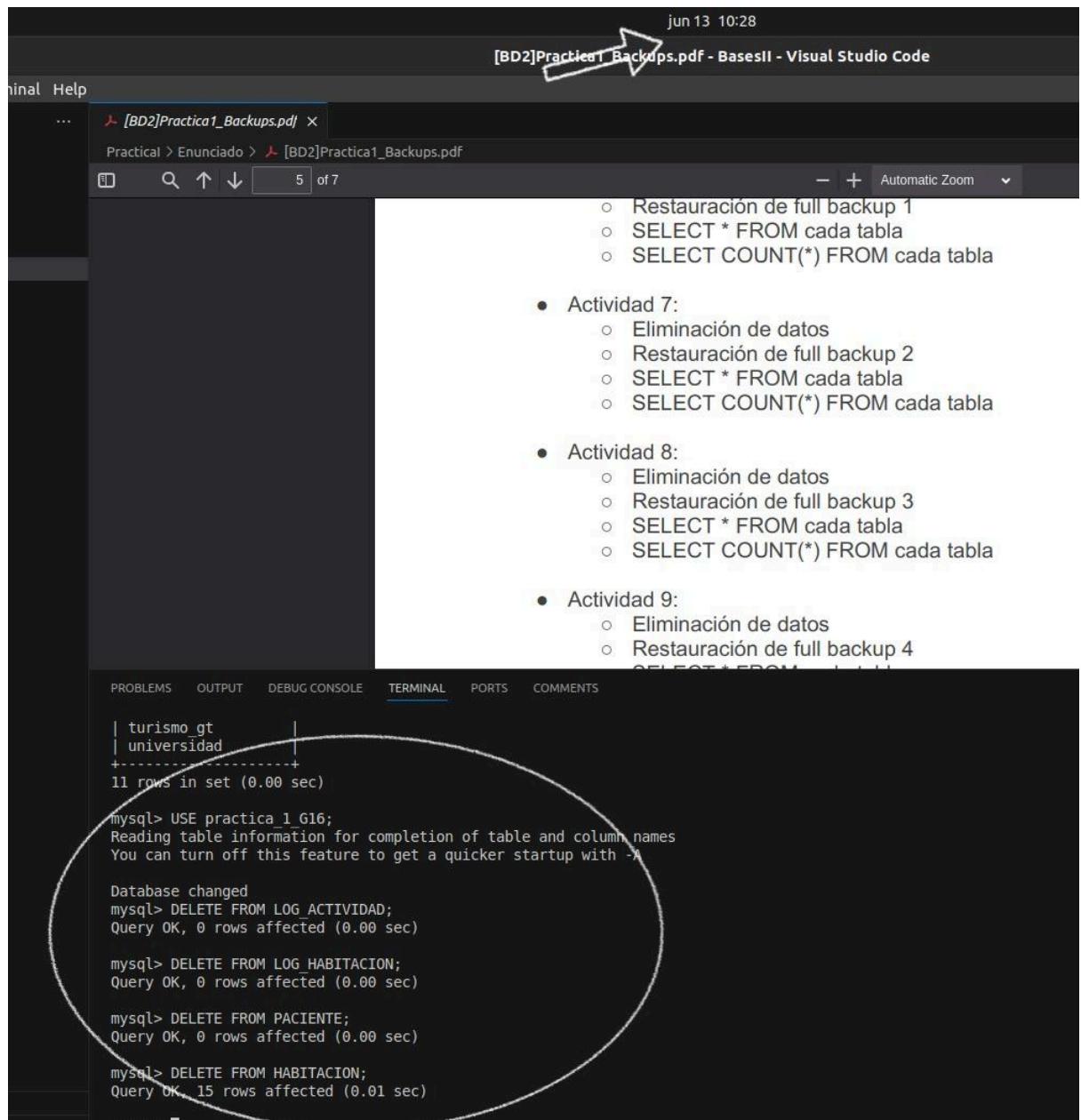
```
1 -- ACTIVIDAD 6
2 • DROP DATABASE practica_1_G16;
3 • SHOW DATABASES;
4
5 • CREATE DATABASE practica_1_G16;
6 -- Actividad 6 Full BackUp 1
7 -- mysql -u root -pDaniel123 practica_1_G16 < /home/danim0/Documents/BasesII/Pra
8
9 • USE practica_1_G16;
```

The results pane below shows a single row of data from the 'LOG_HABITACION' table, with a value of 0.

Imagen 6.10. SELECT COUNT(*) a la tabla LOG_HABITACION.

Actividad 7

Para la segunda actividad correspondiente a la restauración de full backups comenzamos eliminando los datos de nuestra base de datos. Para este caso en particular se decidió eliminar los datos haciendo uso del comando DELETE FROM a cada una de las tablas.



The screenshot shows a Visual Studio Code interface. At the top, there's a PDF document titled "[BD2]Practica1_Backups.pdf - BasesII - Visual Studio Code". Below the PDF, the terminal tab is active, displaying MySQL command-line interface output. A large oval highlights the terminal window, focusing on the deletion of data from a database.

```

jun 13 10:28
[BD2]Practica1_Backups.pdf - BasesII - Visual Studio Code
Terminal Help
... [BD2]Practica1_Backups.pdf x
Practical > Enunciado > [BD2]Practica1_Backups.pdf
5 of 7 Automatic Zoom
○ Restauración de full backup 1
○ SELECT * FROM cada tabla
○ SELECT COUNT(*) FROM cada tabla

● Actividad 7:
○ Eliminación de datos
○ Restauración de full backup 2
○ SELECT * FROM cada tabla
○ SELECT COUNT(*) FROM cada tabla

● Actividad 8:
○ Eliminación de datos
○ Restauración de full backup 3
○ SELECT * FROM cada tabla
○ SELECT COUNT(*) FROM cada tabla

● Actividad 9:
○ Eliminación de datos
○ Restauración de full backup 4
SELECT * FROM cada tabla

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
| turismo gt
| universidad
+-----+
11 rows in set (0.00 sec)

mysql> USE practica_1_G16;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> DELETE FROM LOG_ACTIVIDAD;
Query OK, 0 rows affected (0.00 sec)

mysql> DELETE FROM LOG_HABITACION;
Query OK, 0 rows affected (0.00 sec)

mysql> DELETE FROM PACIENTE;
Query OK, 0 rows affected (0.00 sec)

mysql> DELETE FROM HABITACION;
Query OK, 15 rows affected (0.01 sec)

```

Imagen 7.1. Eliminación de datos en la base de datos.

Después de la eliminación de todos los datos dentro de nuestra base de datos, procedemos a ejecutar el comando para restaurar nuestra base de datos haciendo uso del full backup creado en la Actividad 2.

The screenshot shows a Visual Studio Code interface. At the top, there's a status bar with the date 'jun 13 10:30' and the title '[BD2]Practica1_Backups.pdf - BasesII - Visual Studio Code'. Below the status bar is a dark-themed terminal window. The terminal window has a tab bar with '[BD2]Practica1_Backups.pdf' selected. The main area of the terminal shows a MySQL session:

```

mysql -u root -pDaniel123 practica_1 G16 < /home/danimo/Documents/BasesII/PracticaI/Full_BackUp/Full_BackUp_Act2.sql
mysql: [Warning] Using a password on the command line interface can be insecure.

[Output]

```

Below the terminal, the Visual Studio Code interface includes a navigation bar with 'Terminal' selected, and tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', 'PORTS', and 'COMMENTS'.

Imagen 7.2. Ejecución de comando para restaurar nuestro full backup 2.

Para verificar que se han cargado los datos de manera correcta, ahora ejecutamos el comando SELECT a cada una de nuestras tablas.

The screenshot shows the MySQL Workbench application. At the top, there's a status bar with the date 'jun 13 10:30' and the title 'MySQL Workbench'. Below the status bar is a menu bar with 'File', 'Database', 'Server', 'Tools', 'Scripting', and 'Help'. The main area contains a 'Query 1' editor tab and a 'SQL File 3*' editor tab. The 'Query 1' tab contains the following SQL code:

```

28 -- ACTIVIDAD 7 (SELECT DE TABLAS)
29 • SELECT * FROM HABITACION;
30 • SELECT * FROM PACIENTE;
31 • SELECT * FROM LOG_ACTIVIDAD;
32 • SELECT * FROM LOG_HABITACION;
33
34 -- ACTIVIDAD 7 (SELECT COUNT DE TABLAS)

```

Below the code editor is a results grid titled 'Result Grid' with the following data:

#	idHabitacion	habitacion
1	1	Sala de examenes 1
2	2	Sala de examenes 2
3	3	Sala de examenes 3
4	4	Sala de examenes 4
5	5	Sala de imagenes 1
6	6	Sala de procedimientos 1
7	7	Sala de procedimientos 2
8	8	Sala de procedimientos 3
9	9	Sala de procedimientos 4

Imagen 7.3. SELECT a la tabla HABITACION.

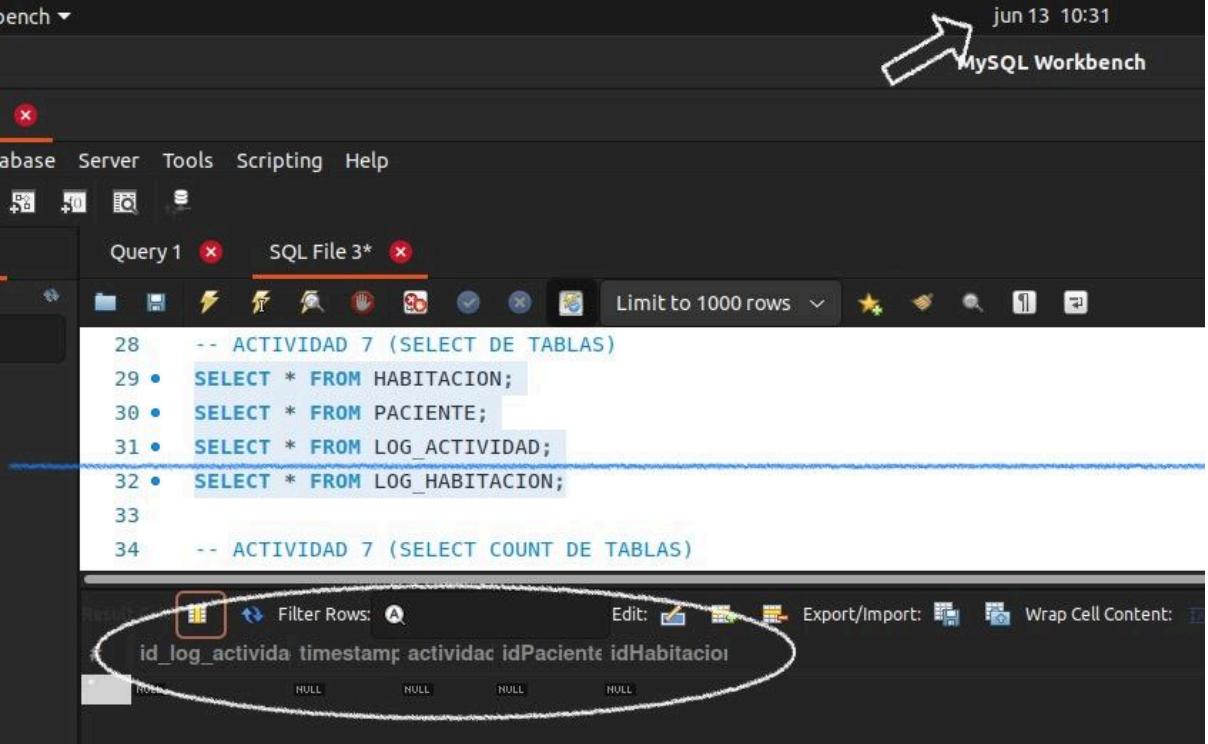
The screenshot shows the MySQL Workbench interface. At the top right, there is a timestamp 'jun 13 10:31' with a large white arrow pointing towards it. The main window has a dark theme. In the center, there is a 'Query 1' tab and a 'SQL File 3*' tab. Below the tabs is a toolbar with various icons. A blue horizontal bar highlights the following SQL code:

```
28 -- ACTIVIDAD 7 (SELECT DE TABLAS)
29 • SELECT * FROM HABITACION;
30 • SELECT * FROM PACIENTE;
31 • SELECT * FROM LOG_ACTIVIDAD;
32 • SELECT * FROM LOG_HABITACION;
33
34 -- ACTIVIDAD 7 (SELECT COUNT DE TABLAS)
```

Below the code, the results are displayed in a table titled 'Result Grid'. The table has columns '#', 'idPaciente', 'edad', and 'genero'. The data is as follows:

#	idPaciente	edad	genero
1	100000	95	Otro
2	100001	40	Femenino
3	100002	42	Masculino
4	100003	8	Femenino
5	100004	88	Masculino
6	100005	23	Masculino
7	100006	60	Femenino
8	100007	49	Femenino
9	100008	44	Femenino
10	100009	55	Femenino
11	100010	28	Femenino
12	100011	13	Femenino
13	100012	62	Masculino
14	100013	32	Masculino

Imagen 7.4. SELECT a la tabla PACIENTE.



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

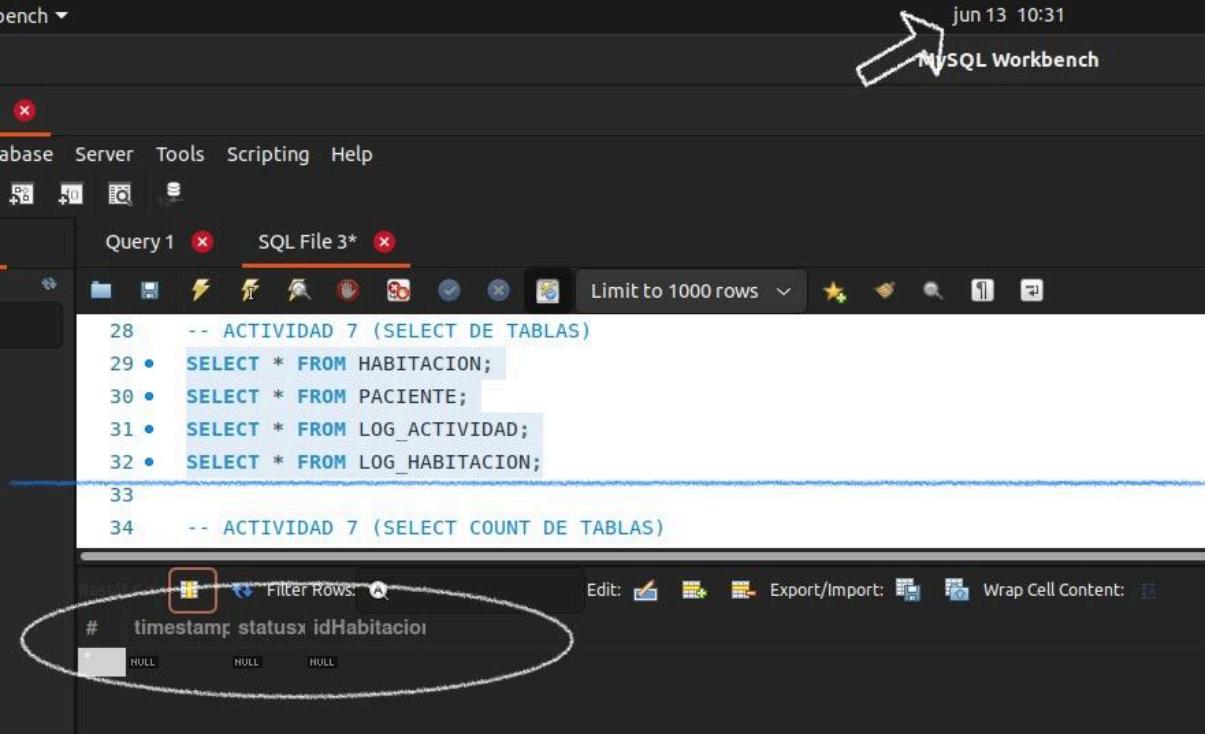
28 -- ACTIVIDAD 7 (SELECT DE TABLAS)
29 • SELECT * FROM HABITACION;
30 • SELECT * FROM PACIENTE;
31 • SELECT * FROM LOG_ACTIVIDAD;
32 • SELECT * FROM LOG_HABITACION;
33
34 -- ACTIVIDAD 7 (SELECT COUNT DE TABLAS)

```

The result set for the 32nd query is displayed below:

	id_log_actividad	timestamp	actividad	idPaciente	idHabitacion
1	NULL	NULL	NULL	NULL	NULL

Imagen 7.5. SELECT a la tabla LOG_ACTIVIDAD.



The screenshot shows the MySQL Workbench interface with a query editor containing the same SQL code as in Image 7.5:

```

28 -- ACTIVIDAD 7 (SELECT DE TABLAS)
29 • SELECT * FROM HABITACION;
30 • SELECT * FROM PACIENTE;
31 • SELECT * FROM LOG_ACTIVIDAD;
32 • SELECT * FROM LOG_HABITACION;
33
34 -- ACTIVIDAD 7 (SELECT COUNT DE TABLAS)

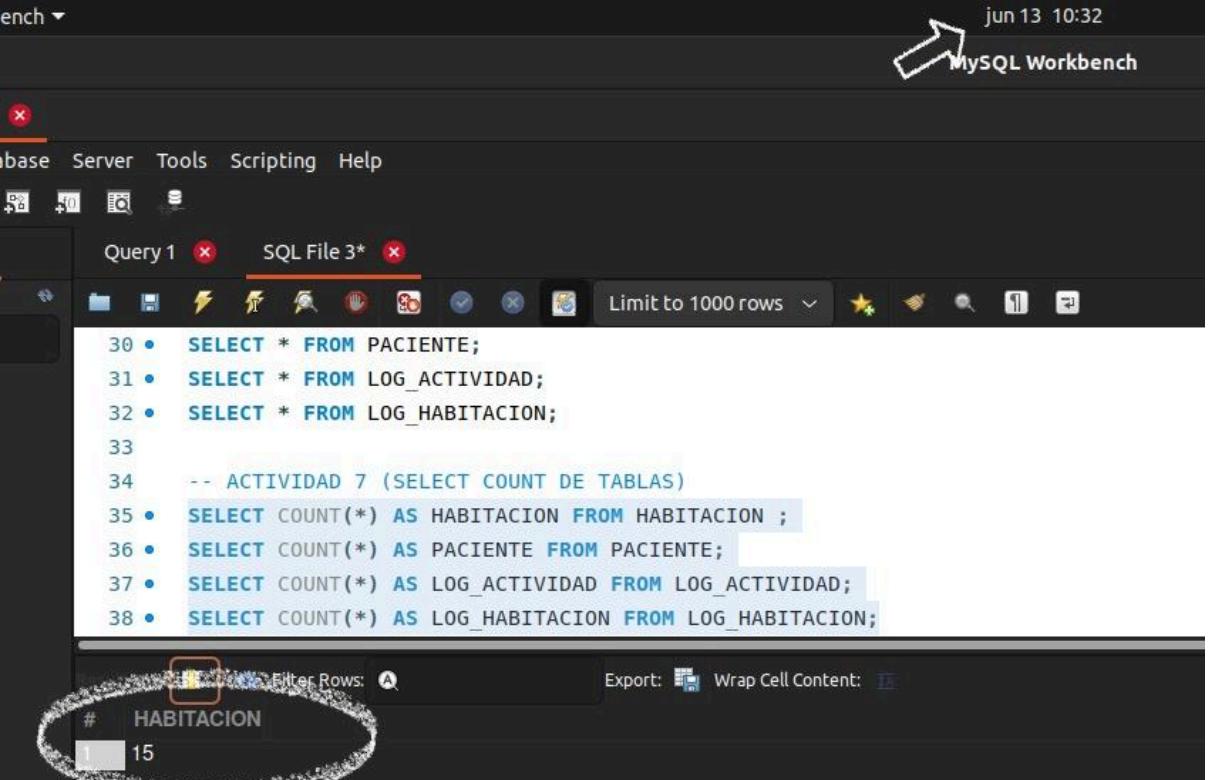
```

The result set for the 32nd query is displayed below:

#	timestamp	status	idHabitacion
1	NULL	NULL	NULL

Imagen 7.6. SELECT a la tabla LOG_HABITACION.

Además de esto, como medida de seguridad ejecutamos el comando SELECT COUNT(*) sobre cada una de nuestras tablas.



The screenshot shows the MySQL Workbench interface with a dark theme. In the top right corner, it displays the date and time: 'jun 13 10:32' and 'MySQL Workbench'. The main window has a toolbar at the top with various icons. Below the toolbar, there are two tabs: 'Query 1' and 'SQL File 3*'. The 'Query 1' tab contains the following SQL code:

```

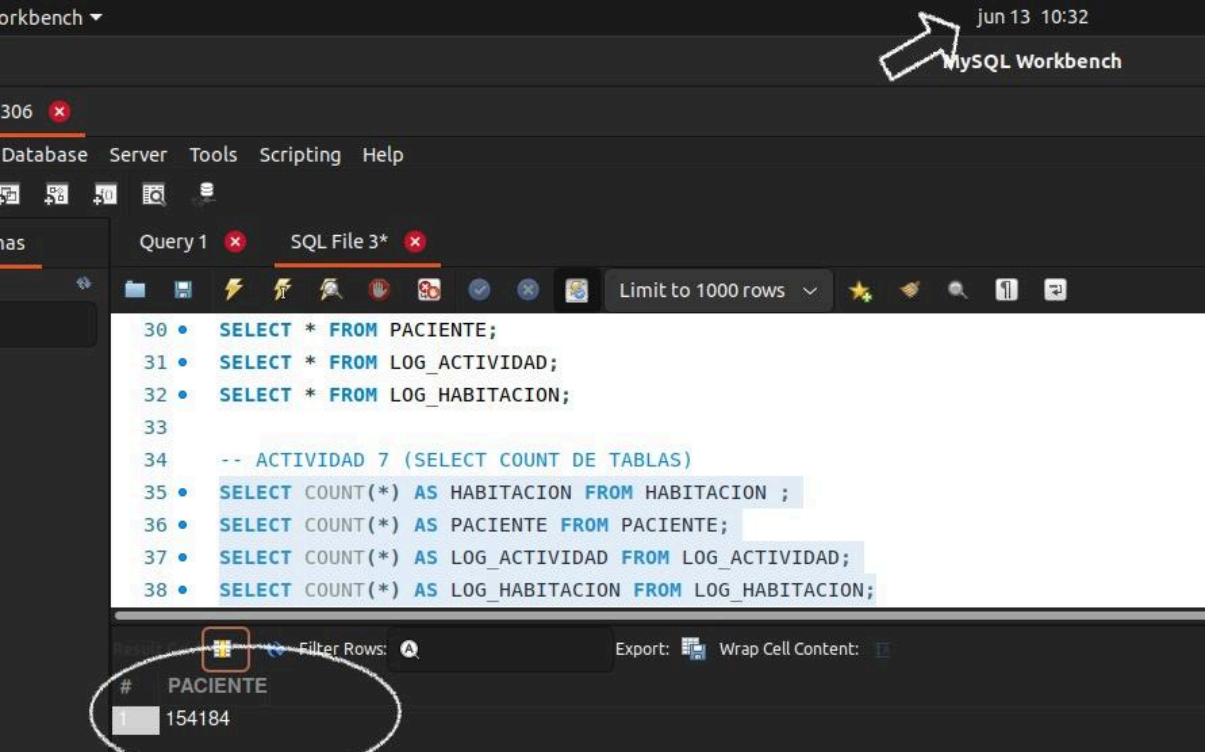
30 •  SELECT * FROM PACIENTE;
31 •  SELECT * FROM LOG_ACTIVIDAD;
32 •  SELECT * FROM LOG_HABITACION;
33
34 -- ACTIVIDAD 7 (SELECT COUNT DE TABLAS)
35 •  SELECT COUNT(*) AS HABITACION FROM HABITACION ;
36 •  SELECT COUNT(*) AS PACIENTE FROM PACIENTE;
37 •  SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD;
38 •  SELECT COUNT(*) AS LOG_HABITACION FROM LOG_HABITACION;

```

Below the code, the results of the last query are displayed in a table:

#	HABITACION
1	15

Imagen 7.7. SELECT COUNT(*) a la tabla HABITACION.



The screenshot shows the MySQL Workbench interface with a dark theme. In the top right corner, it displays the date and time: 'jun 13 10:32' and 'MySQL Workbench'. The main window has a toolbar at the top with various icons. Below the toolbar, there are two tabs: 'Query 1' and 'SQL File 3*'. The 'Query 1' tab contains the same SQL code as in the previous screenshot:

```

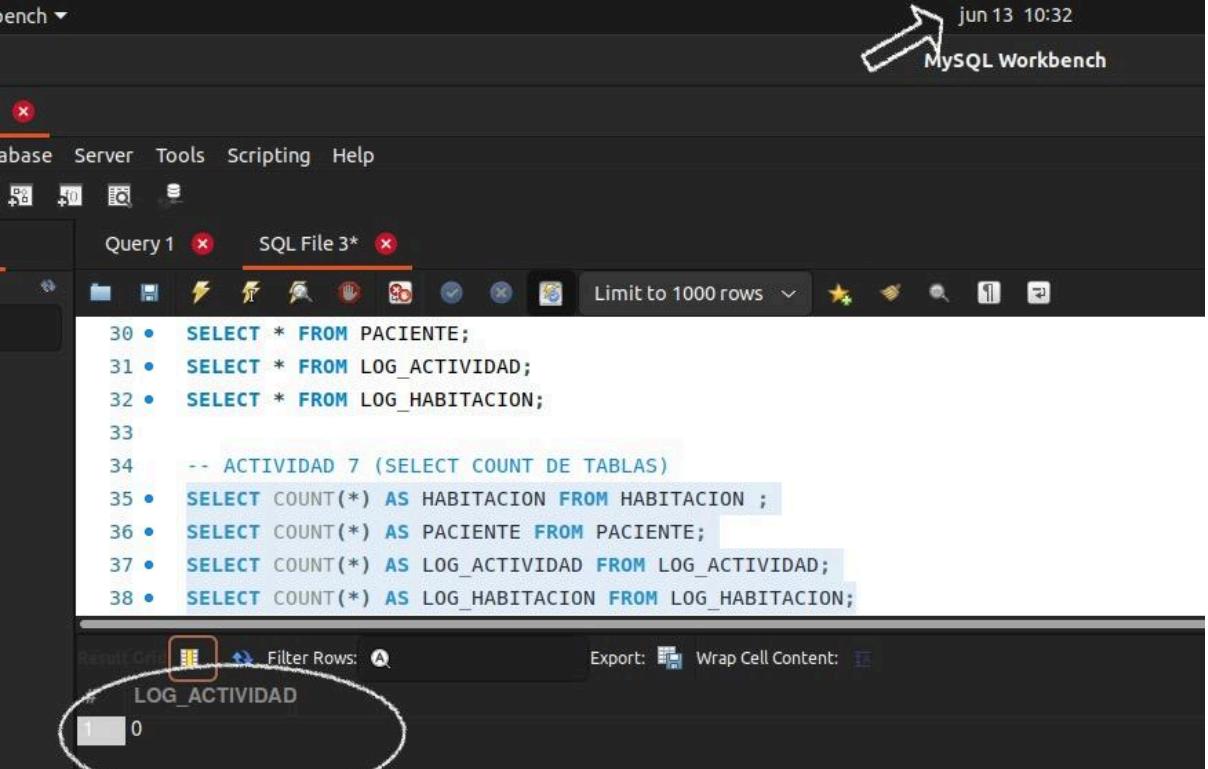
30 •  SELECT * FROM PACIENTE;
31 •  SELECT * FROM LOG_ACTIVIDAD;
32 •  SELECT * FROM LOG_HABITACION;
33
34 -- ACTIVIDAD 7 (SELECT COUNT DE TABLAS)
35 •  SELECT COUNT(*) AS HABITACION FROM HABITACION ;
36 •  SELECT COUNT(*) AS PACIENTE FROM PACIENTE;
37 •  SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD;
38 •  SELECT COUNT(*) AS LOG_HABITACION FROM LOG_HABITACION;

```

Below the code, the results of the last query are displayed in a table:

#	PACIENTE
1	154184

Imagen 7.8. SELECT COUNT(*) a la tabla PACIENTE.



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

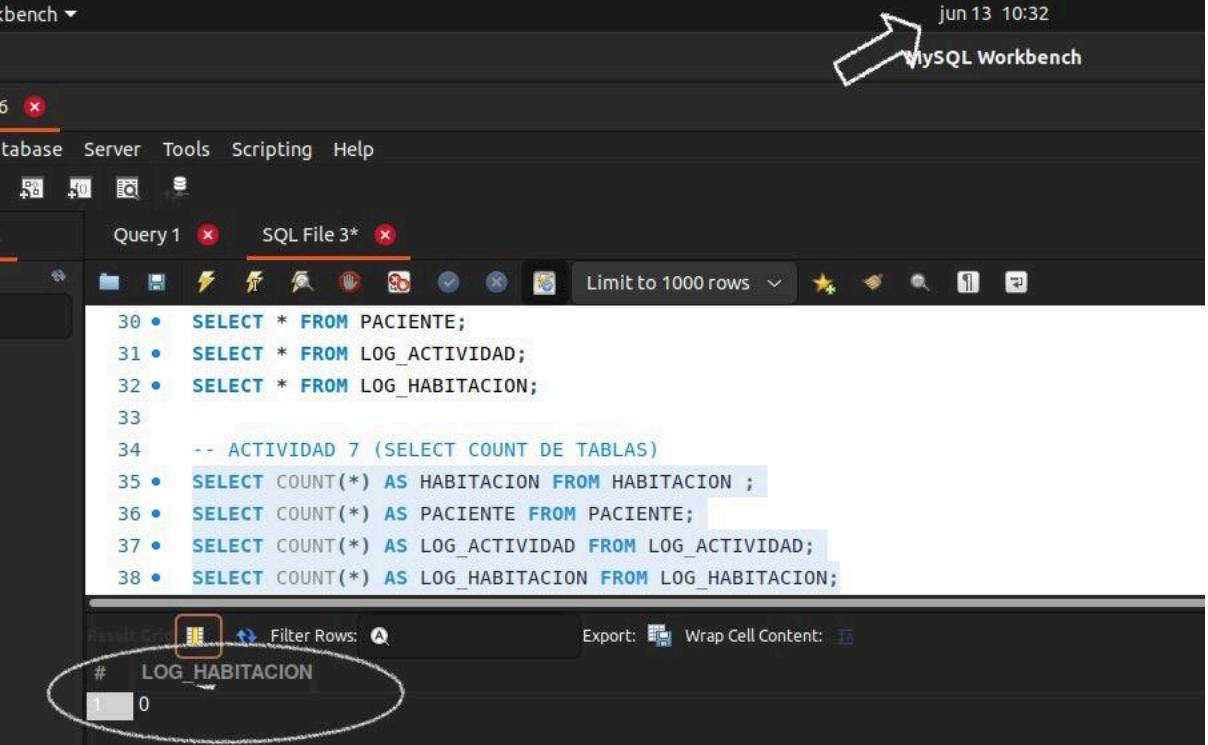
30 • SELECT * FROM PACIENTE;
31 • SELECT * FROM LOG_ACTIVIDAD;
32 • SELECT * FROM LOG_HABITACION;
33
34 -- ACTIVIDAD 7 (SELECT COUNT DE TABLAS)
35 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
36 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE;
37 • SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD;
38 • SELECT COUNT(*) AS LOG_HABITACION FROM LOG_HABITACION;

```

The results pane displays the output for the LOG_ACTIVIDAD table, which has one row with values 1 and 0.

	LOG_ACTIVIDAD
1	0

Imagen 7.9. SELECT COUNT(*) a la tabla LOG_ACTIVIDAD.



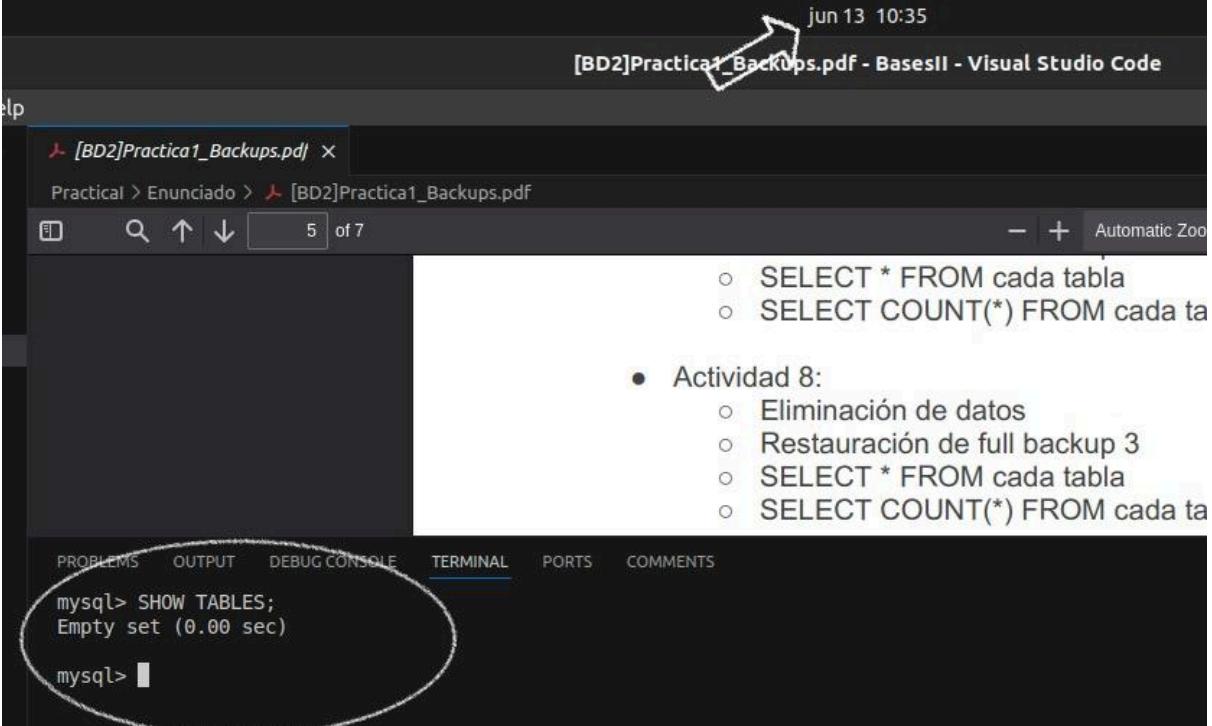
The screenshot shows the MySQL Workbench interface with a query editor containing the same SQL code as in Image 7.9. The results pane displays the output for the LOG_HABITACION table, which has one row with values # and 0.

#	LOG_HABITACION
	0

Imagen 7.10. SELECT COUNT(*) a la tabla LOG_HABITACION.

Actividad 8

Para la tercera actividad correspondiente a la restauración de full backups comenzamos eliminando los datos de nuestra base de datos. Para este caso se decidió eliminar los datos haciendo uso del comando DROP TABLE a cada una de las tablas.



The screenshot shows a Visual Studio Code interface. At the top, there is a PDF document titled "[BD2]Practica1_Backups.pdf". The terminal window below it displays the following MySQL session:

```
mysql> SHOW TABLES;
Empty set (0.00 sec)

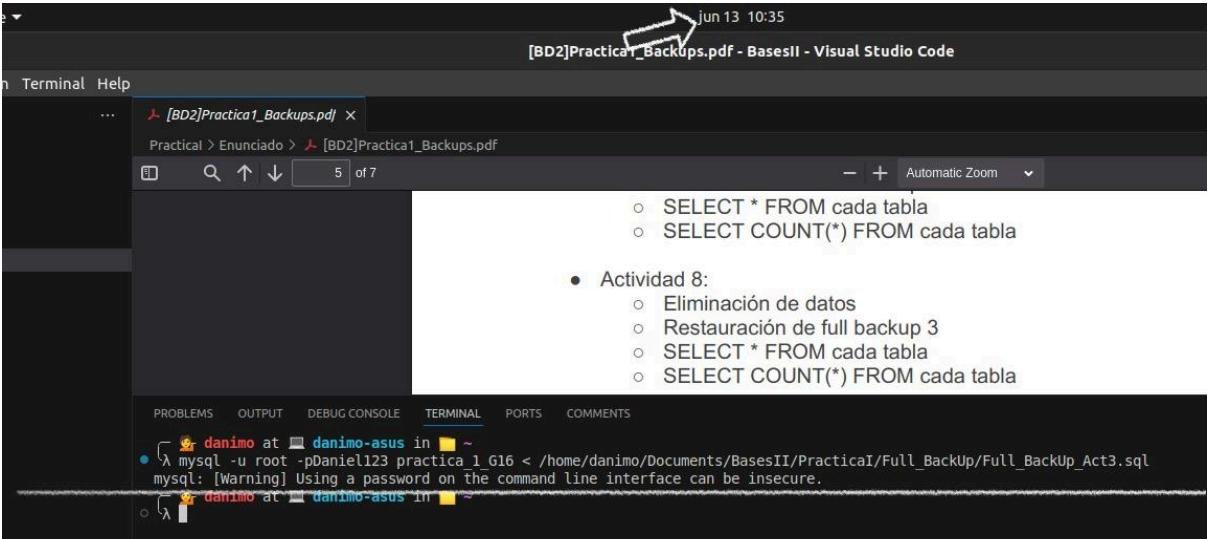
mysql>
```

A white oval highlights the command "SHOW TABLES;". To the right of the terminal, a list of tasks is visible:

- SELECT * FROM cada tabla
- SELECT COUNT(*) FROM cada ta
- Actividad 8:
 - Eliminación de datos
 - Restauración de full backup 3
 - SELECT * FROM cada tabla
 - SELECT COUNT(*) FROM cada ta

Imagen 8.1. Eliminación de tablas en la base de datos.

Después de la eliminación de todas las tablas de nuestra base de datos, procedemos a ejecutar el comando para restaurar nuestra base de datos haciendo uso del full backup creado en la Actividad 3.



The screenshot shows a Visual Studio Code interface. At the top, there is a PDF document titled "[BD2]Practica1_Backups.pdf". The terminal window below it displays the following MySQL session:

```
mysql> mysql -u root -pDaniel123 practica_1 < /home/danimo/Documents/BasesII/PracticaI/Full_BackUp/Full_BackUp_Act3.sql
mysql: [Warning] Using a password on the command line interface can be insecure.

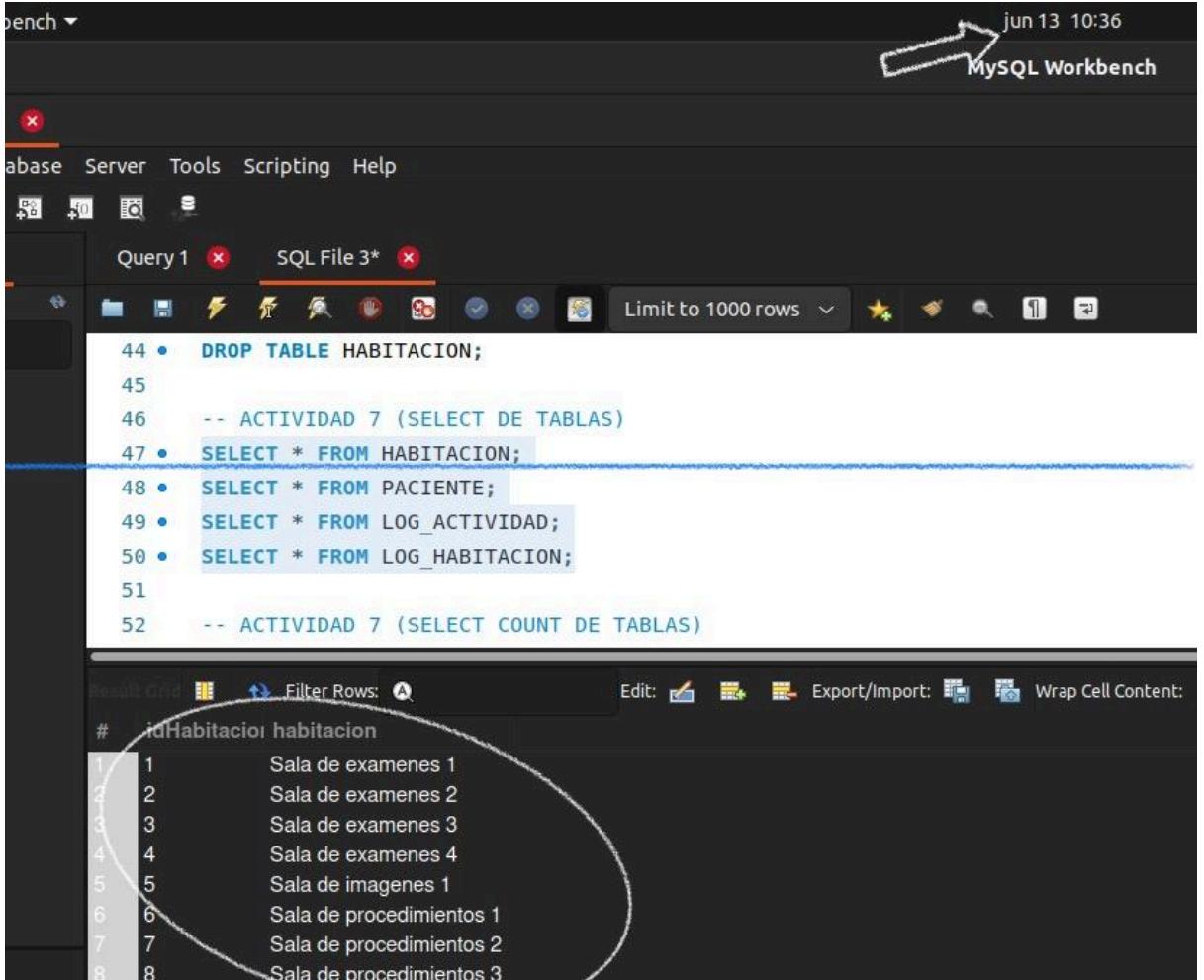
mysql>
```

A white oval highlights the command "mysql -u root -pDaniel123 practica_1 < /home/danimo/Documents/BasesII/PracticaI/Full_BackUp/Full_BackUp_Act3.sql". To the right of the terminal, a list of tasks is visible:

- SELECT * FROM cada tabla
- SELECT COUNT(*) FROM cada tabla
- Actividad 8:
 - Eliminación de datos
 - Restauración de full backup 3
 - SELECT * FROM cada tabla
 - SELECT COUNT(*) FROM cada tabla

Imagen 8.2. Ejecución de comando para restaurar nuestro full backup 3.

Para verificar que se han cargado los datos de manera correcta, ahora ejecutamos el comando SELECT a cada una de nuestras tablas.



The screenshot shows the MySQL Workbench interface. In the top right corner, there is a timestamp "jun 13 10:36" and a red arrow pointing to the text "MySQL Workbench". The main window has a dark theme. At the top, there is a menu bar with "Database", "Server", "Tools", "Scripting", and "Help". Below the menu is a toolbar with various icons. Two tabs are open: "Query 1" and "SQL File 3*". The "Query 1" tab contains the following SQL code:

```
44 • DROP TABLE HABITACION;
45
46 -- ACTIVIDAD 7 (SELECT DE TABLAS)
47 • SELECT * FROM HABITACION;
48 • SELECT * FROM PACIENTE;
49 • SELECT * FROM LOG_ACTIVIDAD;
50 • SELECT * FROM LOG_HABITACION;
51
52 -- ACTIVIDAD 7 (SELECT COUNT DE TABLAS)
```

Below the code, the "Result Grid" pane displays the results of the SELECT query. The table has two columns: "# idHabitacion" and "habitacion". The data is as follows:

#	idHabitacion	habitacion
1	1	Sala de exámenes 1
2	2	Sala de exámenes 2
3	3	Sala de exámenes 3
4	4	Sala de exámenes 4
5	5	Sala de imágenes 1
6	6	Sala de procedimientos 1
7	7	Sala de procedimientos 2
8	8	Sala de procedimientos 3

Imagen 8.3. SELECT a la tabla HABITACION.

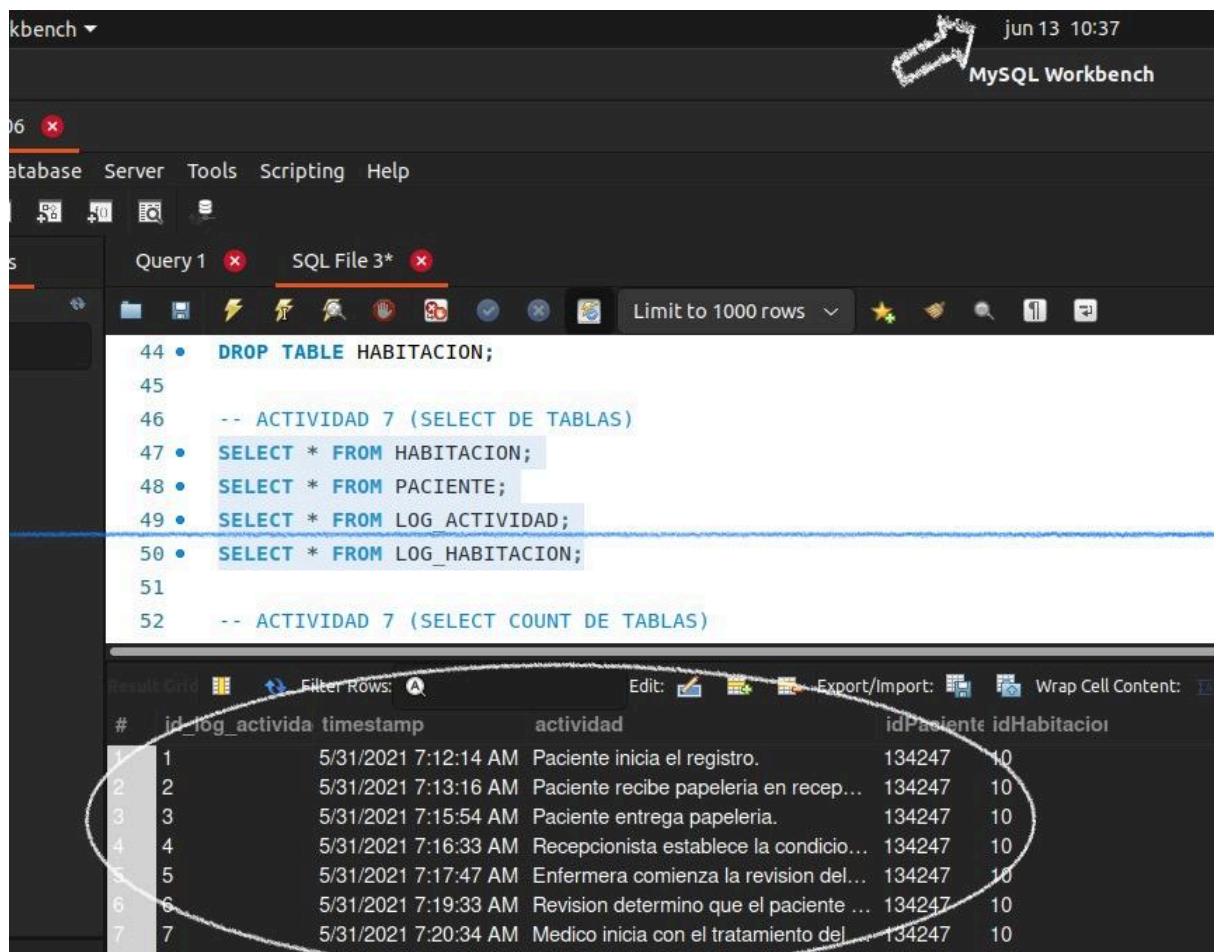
The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a menu bar with Database, Server, Tools, Scripting, and Help. There are two tabs open: "Query 1" and "SQL File 3*". The "Query 1" tab contains the following SQL code:

```
44 • DROP TABLE HABITACION;
45
46 -- ACTIVIDAD 7 (SELECT DE TABLAS)
47 • SELECT * FROM HABITACION;
48 • SELECT * FROM PACIENTE;
49 • SELECT * FROM LOG_ACTIVIDAD;
50 • SELECT * FROM LOG_HABITACION;
51
52 -- ACTIVIDAD 7 (SELECT COUNT DE TABLAS)
```

Below the code, the results pane displays the output of the "SELECT * FROM PACIENTE;" query. The results are presented in a grid with columns: idPaciente, edad, and genero. The data is as follows:

	idPaciente	edad	genero
1	100000	95	Otro
2	100001	40	Femenino
3	100002	42	Masculino
4	100003	8	Femenino
5	100004	88	Masculino
6	100005	23	Masculino
7	100006	60	Femenino
8	100007	49	Femenino

Imagen 8.4. SELECT a la tabla PACIENTE.



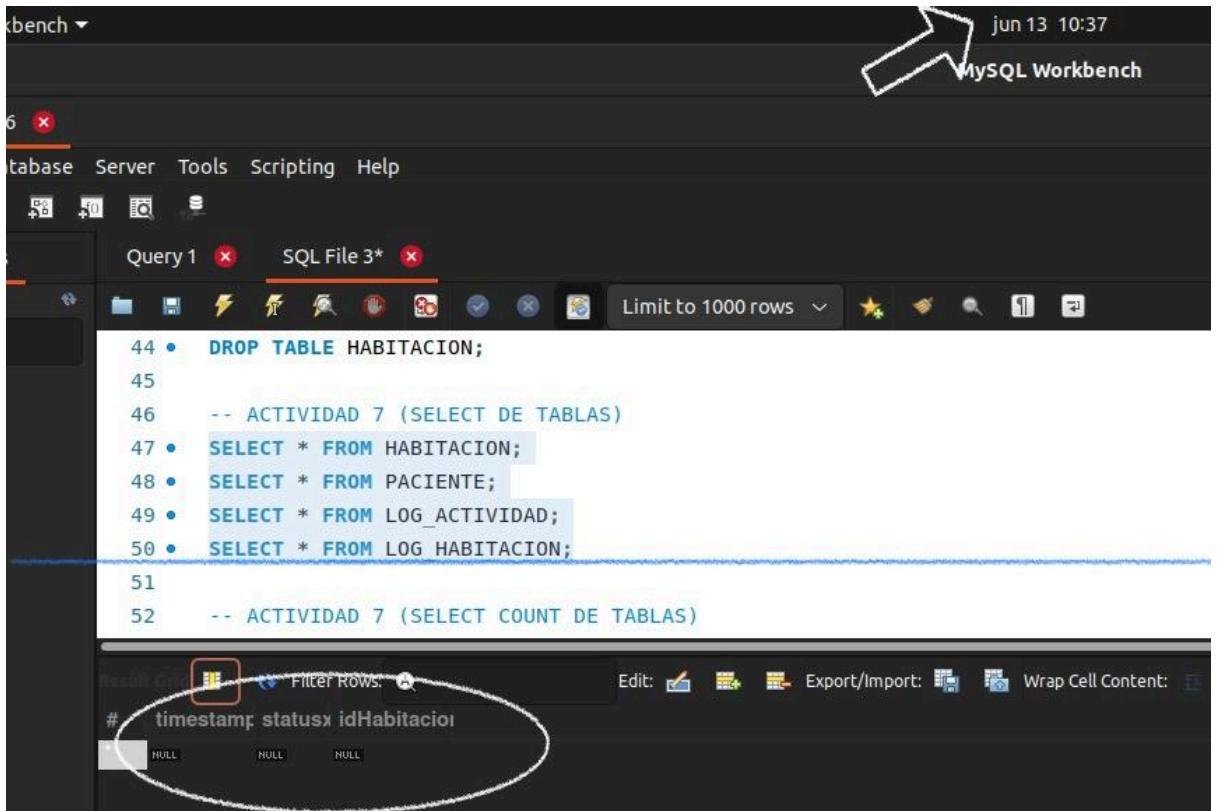
The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a menu bar with Database, Server, Tools, Scripting, and Help. A status bar at the top right indicates "jun 13 10:37". The main area has two tabs: "Query 1" and "SQL File 3*". The "Query 1" tab contains the following SQL code:

```
44 • DROP TABLE HABITACION;
45
46 -- ACTIVIDAD 7 (SELECT DE TABLAS)
47 • SELECT * FROM HABITACION;
48 • SELECT * FROM PACIENTE;
49 • SELECT * FROM LOG ACTIVIDAD;
50 • SELECT * FROM LOG_HABITACION;
51
52 -- ACTIVIDAD 7 (SELECT COUNT DE TABLAS)
```

Below the code is a "Result Grid" showing the output of the last query. The grid has columns: #, id_log_actividad, timestamp, actividad, idPaciente, and idHabitacion. The data is as follows:

#	id_log_actividad	timestamp	actividad	idPaciente	idHabitacion
1	1	5/31/2021 7:12:14 AM	Paciente inicia el registro.	134247	10
2	2	5/31/2021 7:13:16 AM	Paciente recibe papeleria en recepcion...	134247	10
3	3	5/31/2021 7:15:54 AM	Paciente entrega papeleria.	134247	10
4	4	5/31/2021 7:16:33 AM	Recepcionista establece la condicion...	134247	10
5	5	5/31/2021 7:17:47 AM	Enfermera comienza la revision del...	134247	10
6	6	5/31/2021 7:19:33 AM	Revision determino que el paciente ...	134247	10
7	7	5/31/2021 7:20:34 AM	Medico inicia con el tratamiento del...	134247	10

Imagen 8.5. SELECT a la tabla LOG_ACTIVIDAD.



The screenshot shows the MySQL Workbench interface. The title bar says "jun 13 10:37 MySQL Workbench". The menu bar includes "Database", "Server", "Tools", "Scripting", and "Help". The toolbar has various icons for database management. The query editor tab is "Query 1" and the current tab is "SQL File 3*". The SQL code is:

```

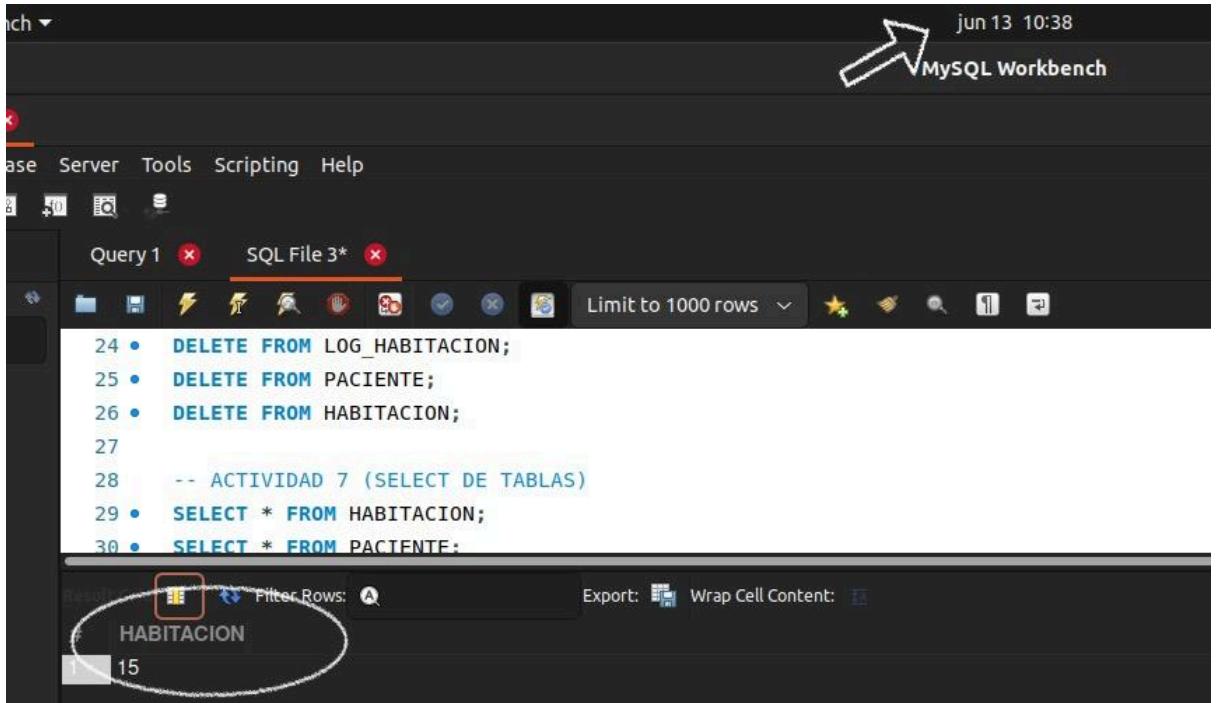
44 • DROP TABLE HABITACION;
45
46 -- ACTIVIDAD 7 (SELECT DE TABLAS)
47 • SELECT * FROM HABITACION;
48 • SELECT * FROM PACIENTE;
49 • SELECT * FROM LOG_ACTIVIDAD;
50 • SELECT * FROM LOG_HABITACION;
51
52 -- ACTIVIDAD 7 (SELECT COUNT DE TABLAS)

```

The results pane shows a table with columns "timestamp", "statusx", and "idHabitacion". The first row contains three NULL values.

Imagen 8.6. SELECT a la tabla LOG_HABITACION.

Además de esto, como medida de seguridad ejecutamos el comando SELECT COUNT(*) sobre cada una de nuestras tablas.



The screenshot shows the MySQL Workbench interface. The title bar says "jun 13 10:38 MySQL Workbench". The menu bar includes "Database", "Server", "Tools", "Scripting", and "Help". The toolbar has various icons for database management. The query editor tab is "Query 1" and the current tab is "SQL File 3*". The SQL code is:

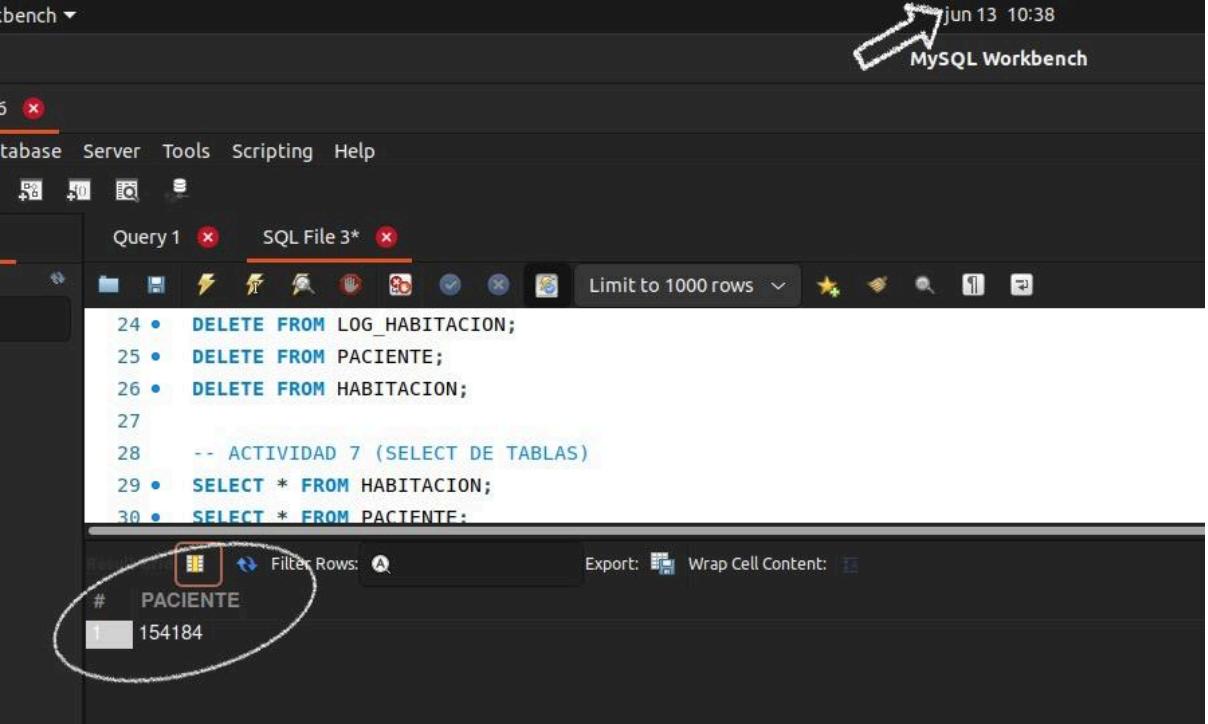
```

24 • DELETE FROM LOG_HABITACION;
25 • DELETE FROM PACIENTE;
26 • DELETE FROM HABITACION;
27
28 -- ACTIVIDAD 7 (SELECT DE TABLAS)
29 • SELECT * FROM HABITACION;
30 • SELECT * FROM PACIENTE;

```

The results pane shows a table with one row labeled "HABITACION" and a value of 15.

Imagen 8.7. SELECT COUNT(*) a la tabla HABITACION.



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

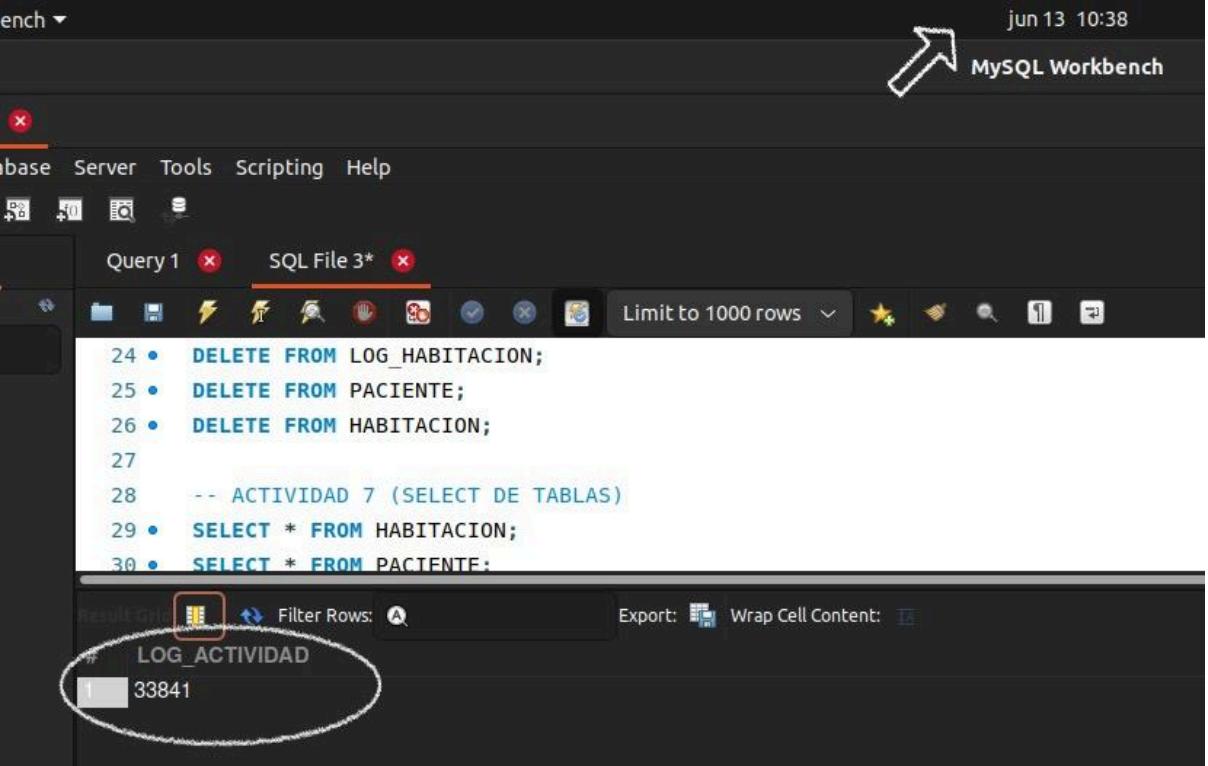
24 • DELETE FROM LOG_HABITACION;
25 • DELETE FROM PACIENTE;
26 • DELETE FROM HABITACION;
27
28 -- ACTIVIDAD 7 (SELECT DE TABLAS)
29 • SELECT * FROM HABITACION;
30 • SELECT * FROM PACIENTE;

```

The results pane displays the output of the last query, which is a single row from the PACIENTE table:

#	PACIENTE
	154184

Imagen 8.8. SELECT COUNT(*) a la tabla PACIENTE.



The screenshot shows the MySQL Workbench interface with a query editor containing the same SQL code as in Image 8.8:

```

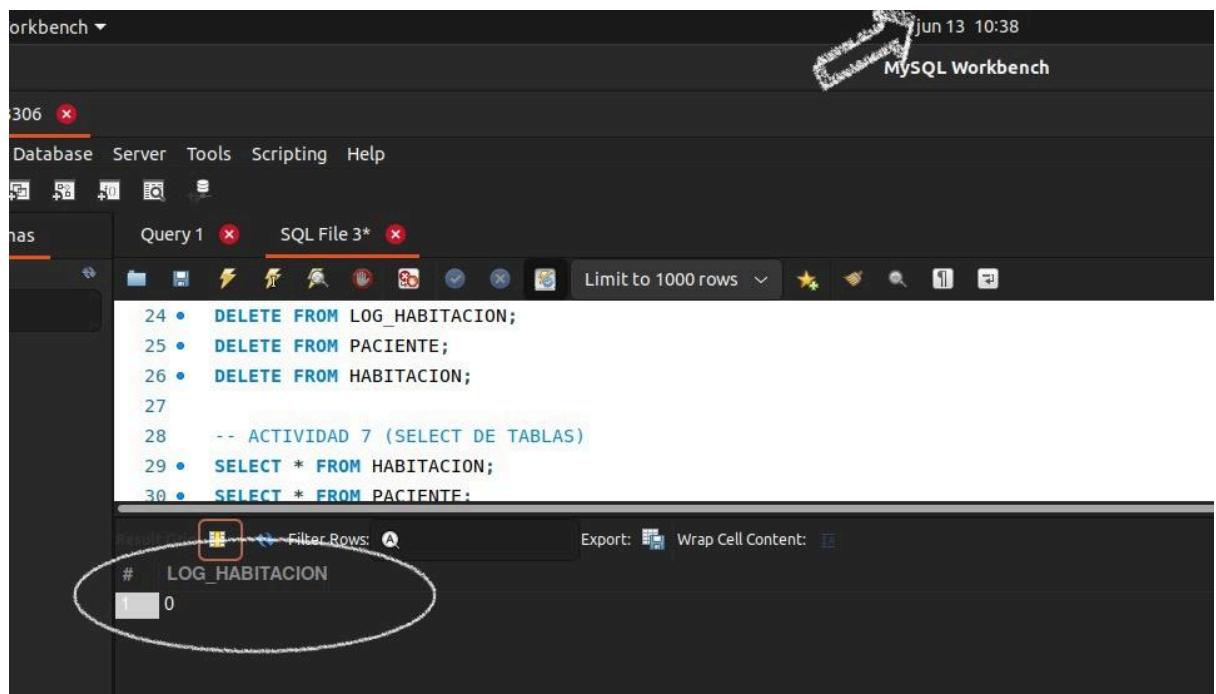
24 • DELETE FROM LOG_HABITACION;
25 • DELETE FROM PACIENTE;
26 • DELETE FROM HABITACION;
27
28 -- ACTIVIDAD 7 (SELECT DE TABLAS)
29 • SELECT * FROM HABITACION;
30 • SELECT * FROM PACIENTE;

```

The results pane displays the output of the last query, which is a single row from the LOG_ACTIVIDAD table:

#	LOG_ACTIVIDAD
	33841

Imagen 8.9. SELECT COUNT(*) a la tabla LOG_ACTIVIDAD.



The screenshot shows the MySQL Workbench interface. In the top right corner, it says "jun 13 10:38" and "MySQL Workbench". The title bar has tabs for "Query 1" and "SQL File 3*". Below the title bar is a toolbar with various icons. The main area contains a SQL editor with the following code:

```
24 • DELETE FROM LOG_HABITACION;
25 • DELETE FROM PACIENTE;
26 • DELETE FROM HABITACION;
27
28 -- ACTIVIDAD 7 (SELECT DE TABLAS)
29 • SELECT * FROM HABITACION;
30 • SELECT * FROM PACIENTE;
```

Below the code, there is a results grid for the table "LOG_HABITACION". The grid has one row with the following data:

#	LOG_HABITACION
1	0

Imagen 8.10. SELECT COUNT(*) a la tabla LOG_HABITACION.

Actividad 9

Para la cuarta actividad correspondiente a la restauración de full backups comenzamos eliminando los datos de nuestra base de datos. Para este caso, al igual que en la actividad anterior, se decidió eliminar los datos haciendo uso del comando DROP TABLE a cada una de las tablas.

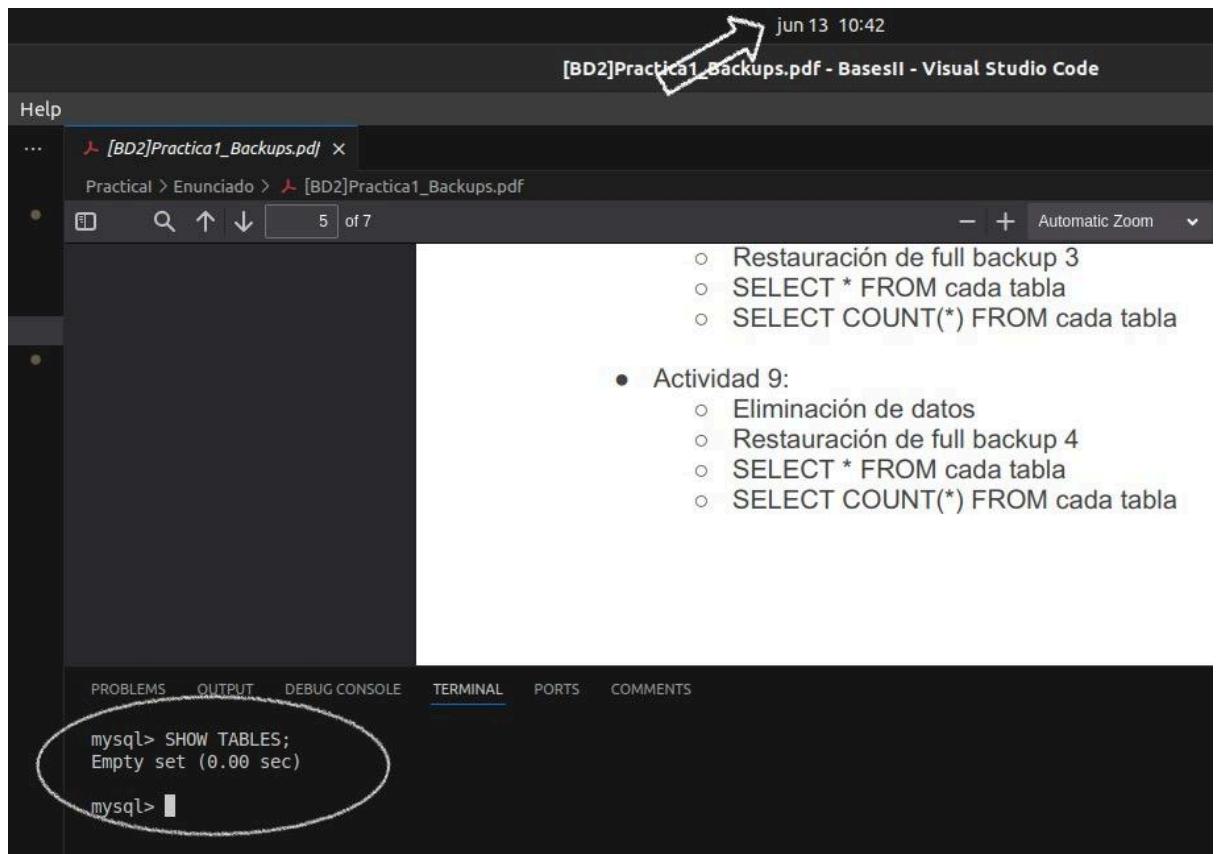


Imagen 9.1. Eliminación de tablas en la base de datos.

Después de la eliminación de todas las tablas de nuestra base de datos, procedemos a ejecutar el comando para restaurar nuestra base de datos haciendo uso del full backup creado en la Actividad 4.

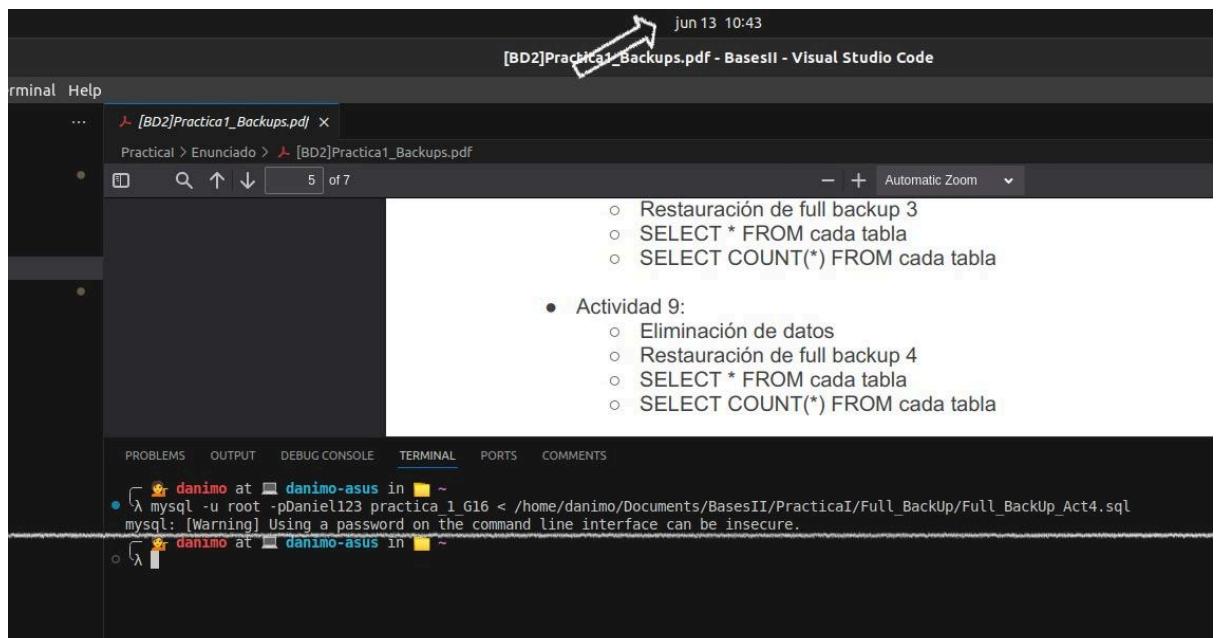
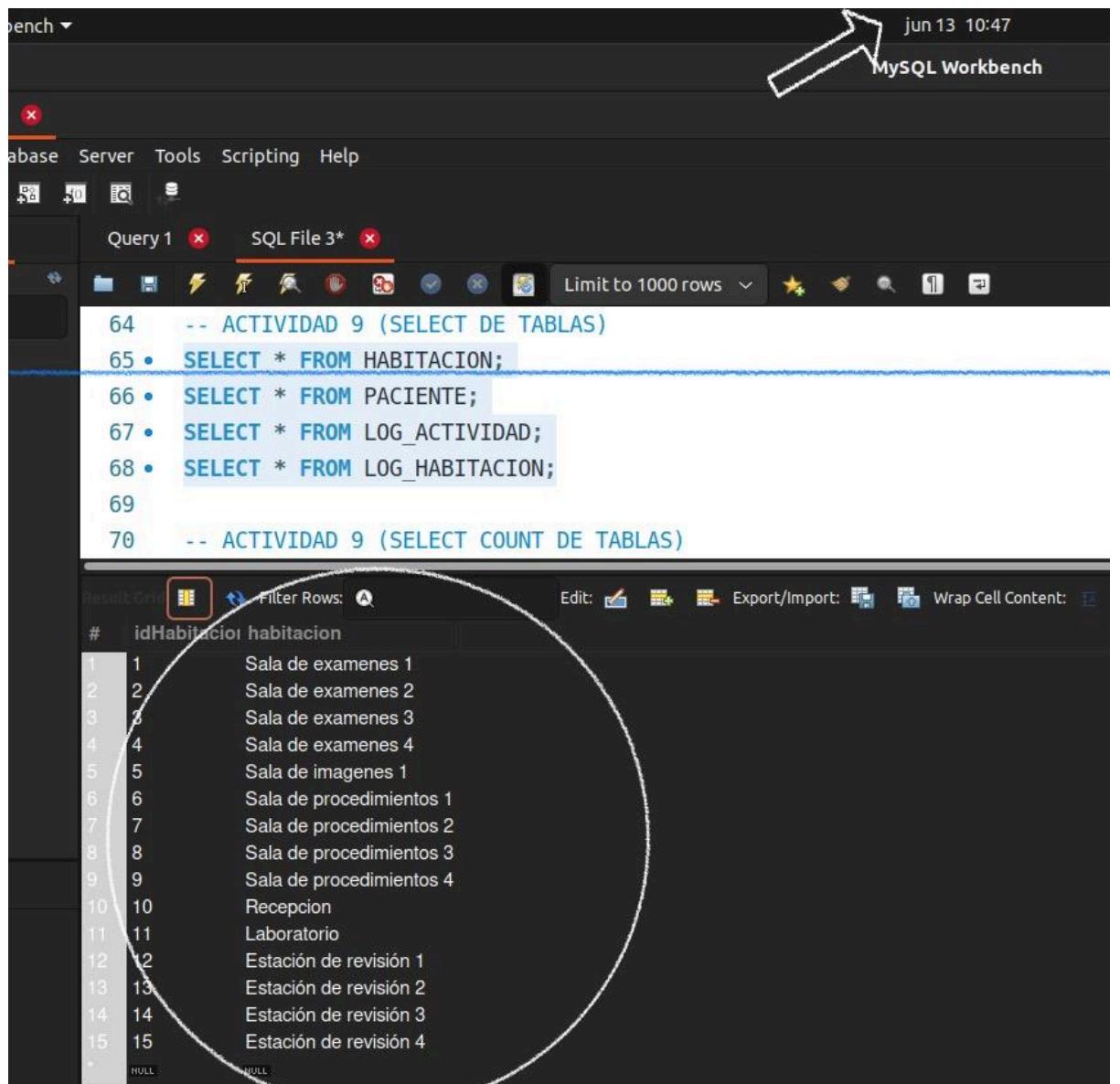


Imagen 9.2. Ejecución de comando para restaurar nuestro full backup 4.

Para verificar que se han cargado los datos de manera correcta, ahora ejecutamos el comando SELECT a cada una de nuestras tablas.



The screenshot shows the MySQL Workbench interface. In the top right corner, there is a timestamp "jun 13 10:47" and the text "MySQL Workbench". The menu bar includes "Database", "Server", "Tools", "Scripting", and "Help". Below the menu is a toolbar with various icons. There are two tabs open: "Query 1" and "SQL File 3*". The "Query 1" tab contains the following SQL code:

```
64 -- ACTIVIDAD 9 (SELECT DE TABLAS)
65 • SELECT * FROM HABITACION;
66 • SELECT * FROM PACIENTE;
67 • SELECT * FROM LOG_ACTIVIDAD;
68 • SELECT * FROM LOG_HABITACION;
69
70 -- ACTIVIDAD 9 (SELECT COUNT DE TABLAS)
```

The results pane shows the output of the SELECT * FROM HABITACION; query. The table has two columns: "# idHabitacion" and "habitacion". The data is as follows:

#	idHabitacion	habitacion
1	1	Sala de exámenes 1
2	2	Sala de exámenes 2
3	3	Sala de exámenes 3
4	4	Sala de exámenes 4
5	5	Sala de imágenes 1
6	6	Sala de procedimientos 1
7	7	Sala de procedimientos 2
8	8	Sala de procedimientos 3
9	9	Sala de procedimientos 4
10	10	Recepción
11	11	Laboratorio
12	12	Estación de revisión 1
13	13	Estación de revisión 2
14	14	Estación de revisión 3
15	15	Estación de revisión 4
*		NULL

Imagen 9.3. SELECT a la tabla HABITACION.

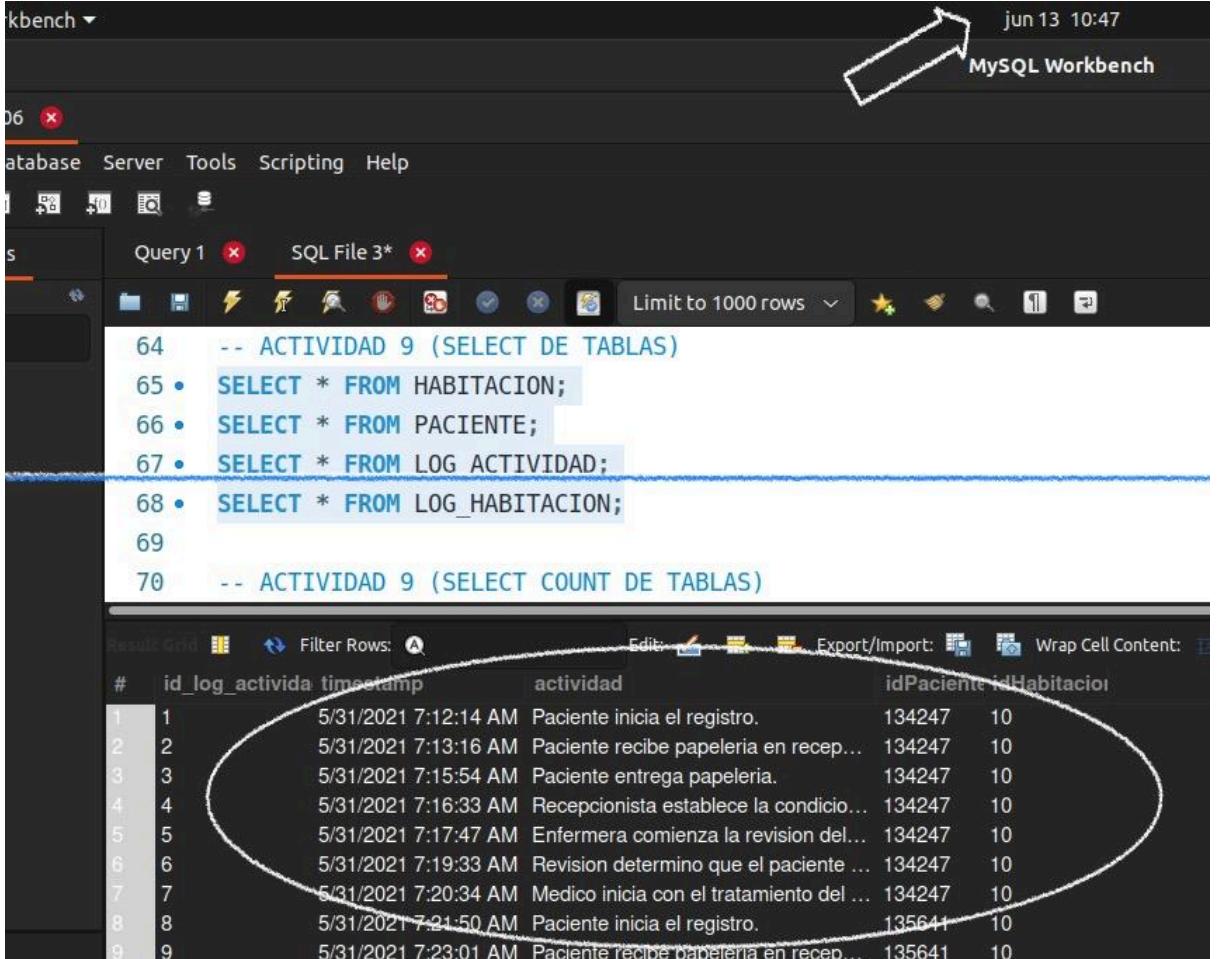
The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a menu bar with Database, Server, Tools, Scripting, and Help. There are two tabs open: 'Query 1' and 'SQL File 3*'. The SQL editor contains the following code:

```
64 -- ACTIVIDAD 9 (SELECT DE TABLAS)
65 • SELECT * FROM HABITACION;
66 • SELECT * FROM PACIENTE;
67 • SELECT * FROM LOG_ACTIVIDAD;
68 • SELECT * FROM LOG_HABITACION;
69
70 -- ACTIVIDAD 9 (SELECT COUNT DE TABLAS)
```

Below the code, the results pane displays a table with the following data:

#	idPaciente	edad	genero
1	100000	95	Otro
2	100001	40	Femenino
3	100002	42	Masculino
4	100003	8	Femenino
5	100004	88	Masculino
6	100005	23	Masculino
7	100006	60	Femenino
8	100007	49	Femenino
9	100008	44	Femenino
10	100009	55	Femenino
11	100010	28	Femenino
12	100011	13	Femenino

Imagen 9.4. SELECT a la tabla PACIENTE.



jun 13 10:47
MySQL Workbench

06

atabase Server Tools Scripting Help

Query 1 SQL File 3*

```

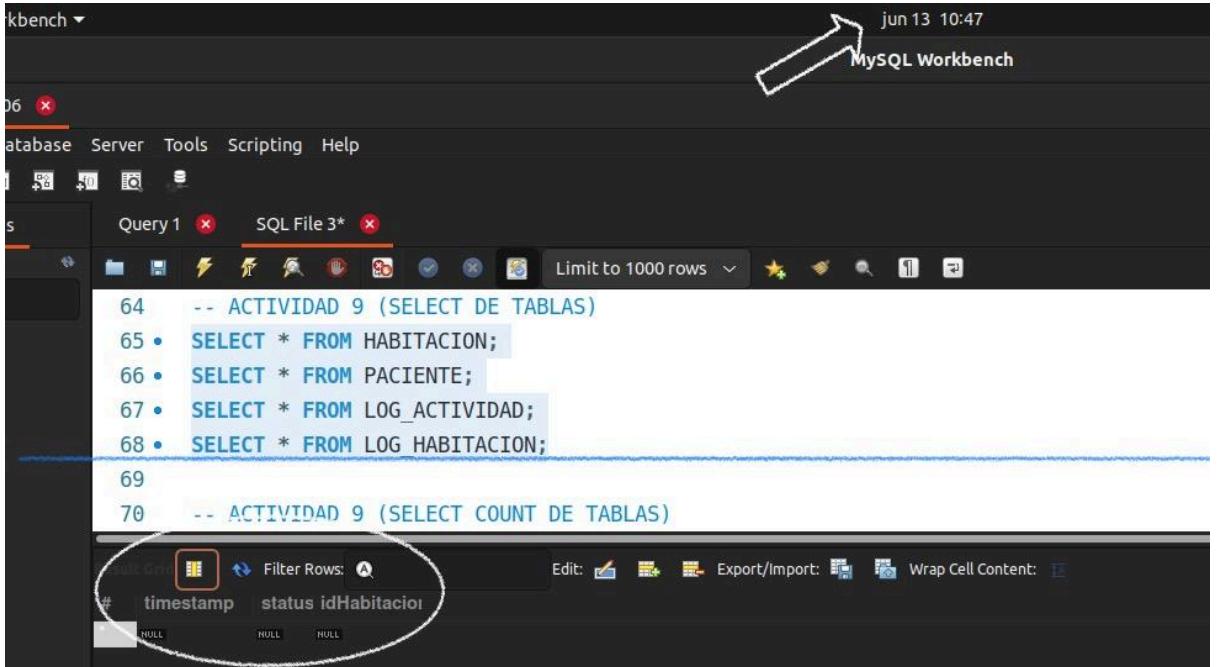
64 -- ACTIVIDAD 9 (SELECT DE TABLAS)
65 • SELECT * FROM HABITACION;
66 • SELECT * FROM PACIENTE;
67 • SELECT * FROM LOG ACTIVIDAD;
68 • SELECT * FROM LOG_HABITACION;
69
70 -- ACTIVIDAD 9 (SELECT COUNT DE TABLAS)

```

Result Grid Filter Rows: A Edit: Export/Import: Wrap Cell Content: A

#	id_log_actividad	timestamp	actividad	idPaciente	idHabitacion
1	1	5/31/2021 7:12:14 AM	Paciente inicia el registro.	134247	10
2	2	5/31/2021 7:13:16 AM	Paciente recibe papeleria en recepcion.	134247	10
3	3	5/31/2021 7:15:54 AM	Paciente entrega papeleria.	134247	10
4	4	5/31/2021 7:16:33 AM	Recepcionista establece la condicion de paciente.	134247	10
5	5	5/31/2021 7:17:47 AM	Enfermera comienza la revision del paciente.	134247	10
6	6	5/31/2021 7:19:33 AM	Revision determino que el paciente tiene fiebre.	134247	10
7	7	5/31/2021 7:20:34 AM	Medico inicia con el tratamiento del paciente.	134247	10
8	8	5/31/2021 7:21:50 AM	Paciente inicia el registro.	135641	10
9	9	5/31/2021 7:23:01 AM	Paciente recibe papeleria en recepcion.	135641	10

Imagen 9.5. SELECT a la tabla LOG_ACTIVIDAD.



jun 13 10:47
MySQL Workbench

06

atabase Server Tools Scripting Help

Query 1 SQL File 3*

```

64 -- ACTIVIDAD 9 (SELECT DE TABLAS)
65 • SELECT * FROM HABITACION;
66 • SELECT * FROM PACIENTE;
67 • SELECT * FROM LOG ACTIVIDAD;
68 • SELECT * FROM LOG_HABITACION;
69
70 -- ACTIVIDAD 9 (SELECT COUNT DE TABLAS)

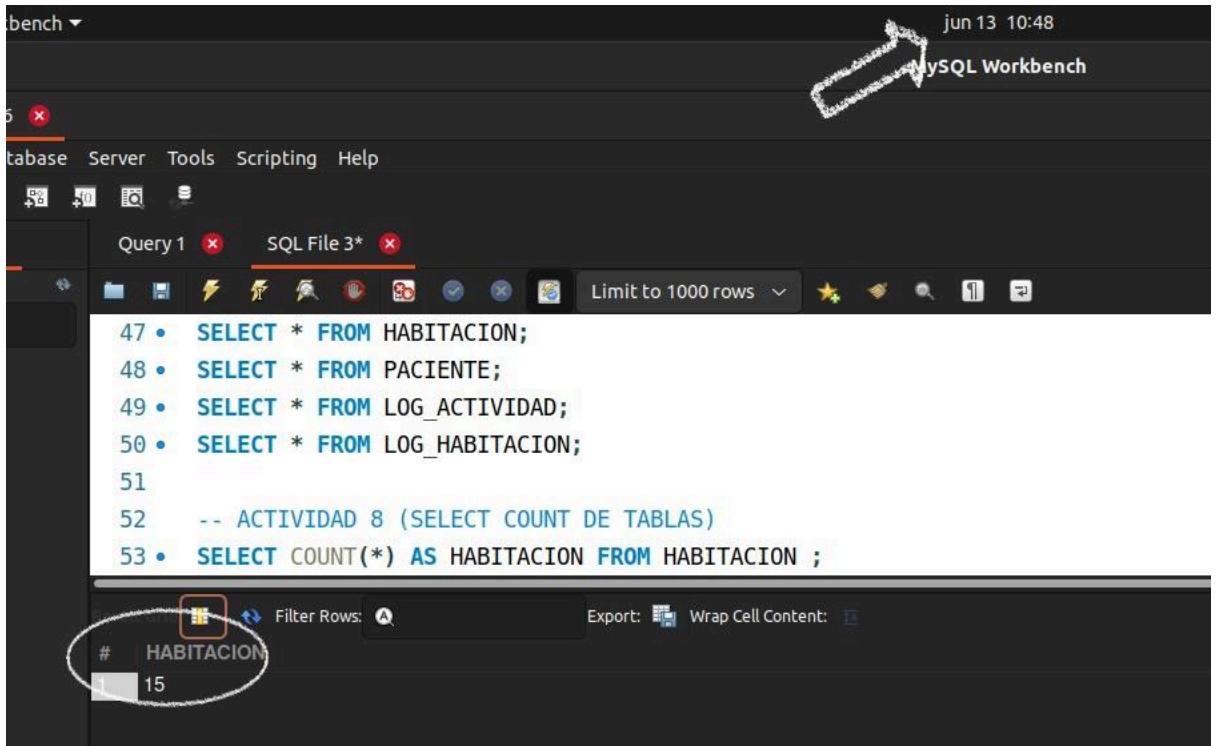
```

Result Grid Filter Rows: A Edit: Export/Import: Wrap Cell Content: A

#	timestamp	status	idHabitacion
HULL	HULL	HULL	

Imagen 9.6. SELECT a la tabla LOG_HABITACION.

Además de esto, como medida de seguridad ejecutamos el comando SELECT COUNT(*) sobre cada una de nuestras tablas.



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

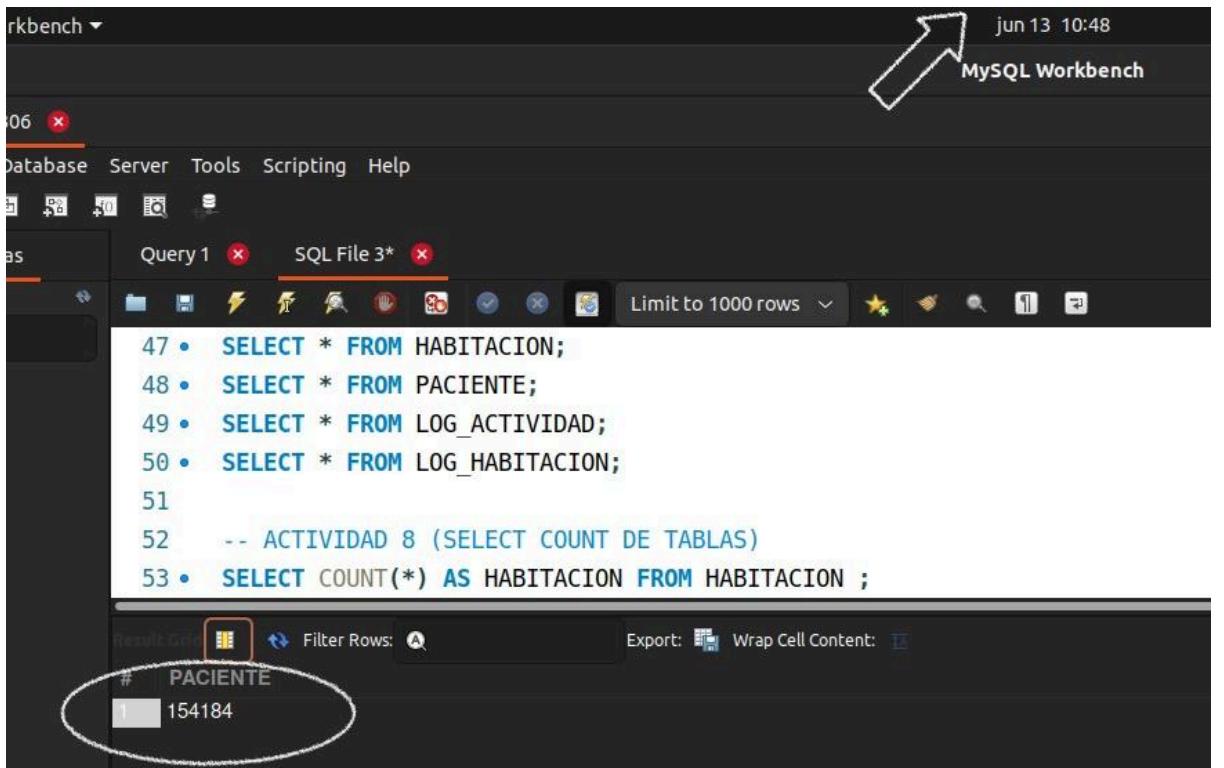
47 •  SELECT * FROM HABITACION;
48 •  SELECT * FROM PACIENTE;
49 •  SELECT * FROM LOG_ACTIVIDAD;
50 •  SELECT * FROM LOG_HABITACION;
51
52 -- ACTIVIDAD 8 (SELECT COUNT DE TABLAS)
53 •  SELECT COUNT(*) AS HABITACION FROM HABITACION ;

```

The results pane displays the output for the last query:

#	HABITACION
1	15

Imagen 9.7. SELECT COUNT(*) a la tabla HABITACION.



The screenshot shows the MySQL Workbench interface with a query editor containing the same SQL code as in Image 9.7:

```

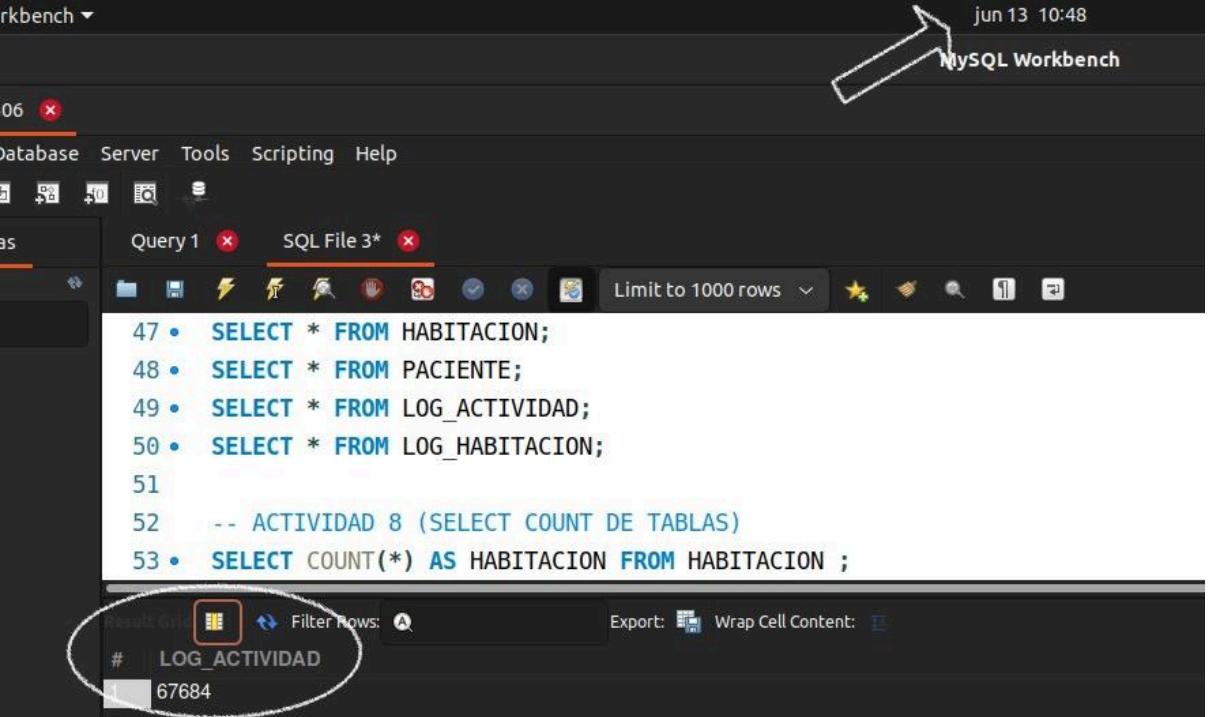
47 •  SELECT * FROM HABITACION;
48 •  SELECT * FROM PACIENTE;
49 •  SELECT * FROM LOG_ACTIVIDAD;
50 •  SELECT * FROM LOG_HABITACION;
51
52 -- ACTIVIDAD 8 (SELECT COUNT DE TABLAS)
53 •  SELECT COUNT(*) AS HABITACION FROM HABITACION ;

```

The results pane displays the output for the last query:

#	PACIENTE
1	154184

Imagen 9.8. SELECT COUNT(*) a la tabla PACIENTE.



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

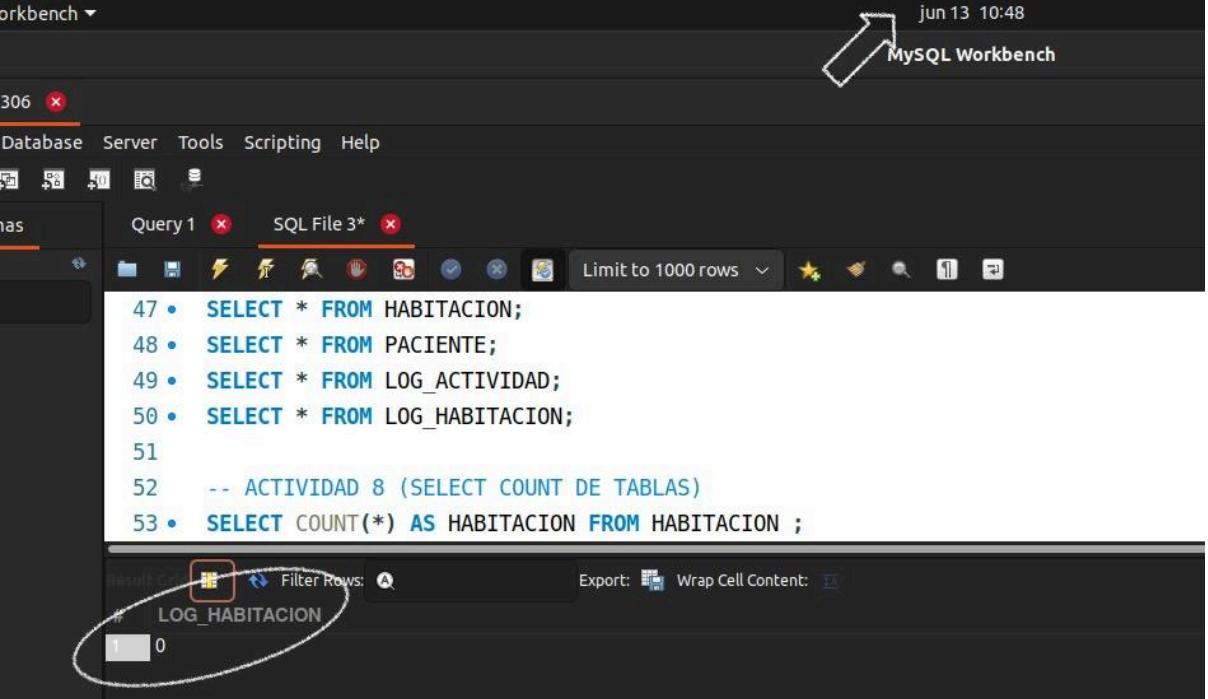
47 •  SELECT * FROM HABITACION;
48 •  SELECT * FROM PACIENTE;
49 •  SELECT * FROM LOG_ACTIVIDAD;
50 •  SELECT * FROM LOG_HABITACION;
51
52 -- ACTIVIDAD 8 (SELECT COUNT DE TABLAS)
53 •  SELECT COUNT(*) AS HABITACION FROM HABITACION ;

```

The results pane displays the count for the LOG_ACTIVIDAD table:

#	LOG_ACTIVIDAD
1	67684

Imagen 9.9. SELECT COUNT(*) a la tabla LOG_ACTIVIDAD.



The screenshot shows the MySQL Workbench interface with a query editor containing the same SQL code as in Image 9.9:

```

47 •  SELECT * FROM HABITACION;
48 •  SELECT * FROM PACIENTE;
49 •  SELECT * FROM LOG_ACTIVIDAD;
50 •  SELECT * FROM LOG_HABITACION;
51
52 -- ACTIVIDAD 8 (SELECT COUNT DE TABLAS)
53 •  SELECT COUNT(*) AS HABITACION FROM HABITACION ;

```

The results pane displays the count for the LOG_HABITACION table:

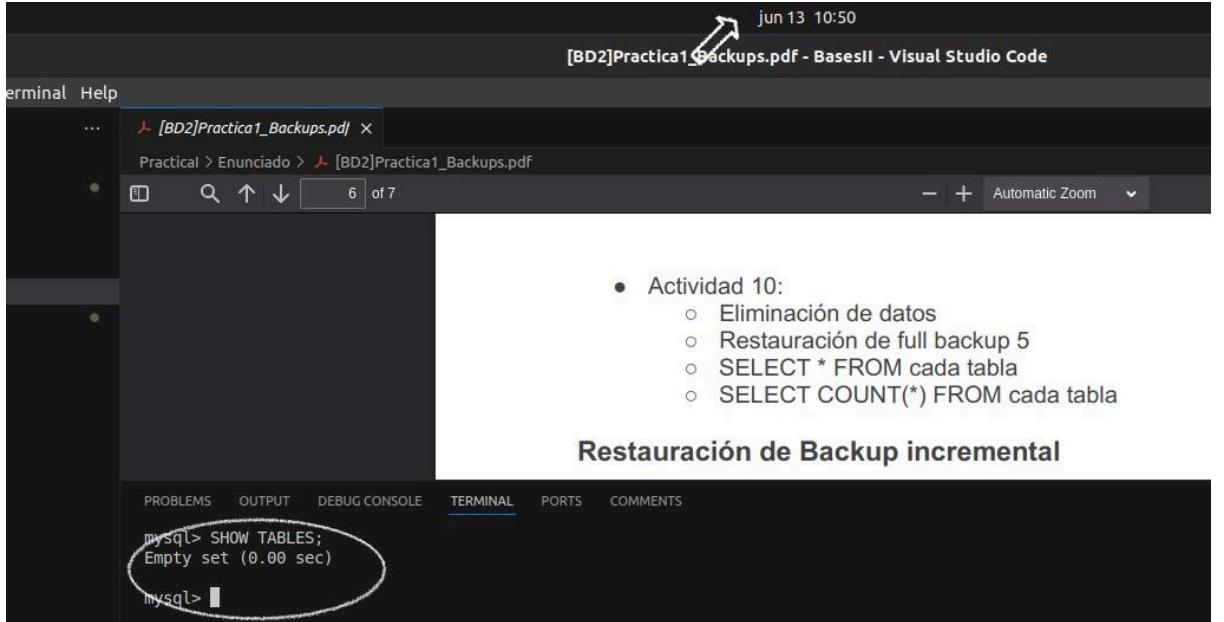
#	LOG_HABITACION
1	0

Imagen 9.10. SELECT COUNT(*) a la tabla LOG_HABITACION.

Actividad 10

Para la última actividad correspondiente a la restauración de full backups comenzamos eliminando los datos de nuestra base de datos. Para este caso, al igual que en la

actividad anterior, se decidió eliminar los datos haciendo uso del comando `DROP TABLE` a cada una de las tablas.



The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal tab is selected at the bottom. Inside the terminal, the MySQL command `SHOW TABLES;` is run, resulting in an empty set. A red oval highlights this command. Above the terminal, a PDF document titled "Practica1_Backups.pdf" is visible, containing a list of activities. One activity is highlighted with a red arrow pointing to it.

```
mysql> SHOW TABLES;
Empty set (0.00 sec)

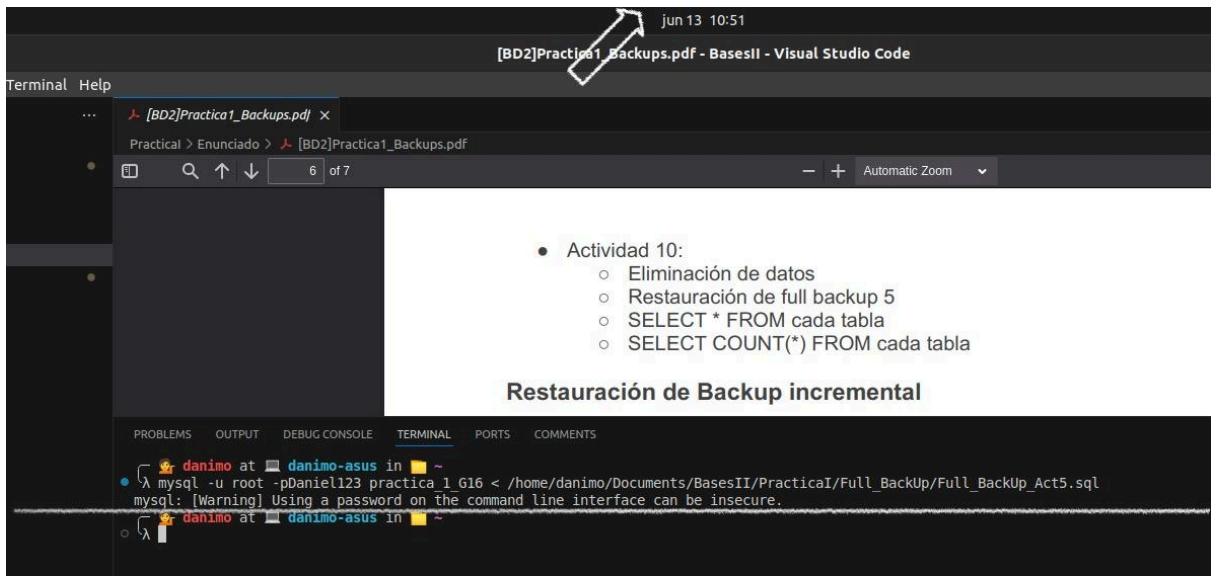
mysql>
```

- Actividad 10:
 - Eliminación de datos
 - Restauración de full backup 5
 - `SELECT * FROM` cada tabla
 - `SELECT COUNT(*) FROM` cada tabla

Restauración de Backup incremental

Imagen 10.1. Eliminación de tablas en la base de datos.

Después de la eliminación de todas las tablas de nuestra base de datos, procedemos a ejecutar el comando para restaurar nuestra base de datos haciendo uso del full backup creado en la Actividad 4.



The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal tab is selected at the bottom. A MySQL command is being run to restore a full backup from a file named `Full_BackUp_Act5.sql`. The command starts with `mysql -u root -p`. A red oval highlights the password part of the command. Above the terminal, the same PDF document from Image 10.1 is visible, with the same highlighted activity.

```
danimo at danimo-asus ~
mysql -u root -pDaniel123 practica1 G16 < /home/danimo/Documents/BasesII/Practica1/Full_BackUp/Full_BackUp_Act5.sql
mysql: [Warning] Using a password on the command line interface can be insecure.

danimo at danimo-asus ~
```

Imagen 10.2. Ejecución de comando para restaurar nuestro full backup 5.

Para verificar que se han cargado los datos de manera correcta, ahora ejecutamos el comando `SELECT` a cada una de nuestras tablas.

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a menu bar with Database, Server, Tools, Scripting, and Help. A status bar at the top right shows "jun 13 10:51" and "MySQL Workbench". The main area has two tabs: "Query 1" and "SQL File 3*". The SQL tab contains the following code:

```
82 -- ACTIVIDAD 10 (SELECT DE TABLAS)
83 • SELECT * FROM HABITACION;
84 • SELECT * FROM PACIENTE;
85 • SELECT * FROM LOG_ACTIVIDAD;
86 • SELECT * FROM LOG_HABITACION;
87
```

Below the code is a result grid titled "Result Grid". It has a header row with columns "#", "idHabitacion" (with a red circle around it), and "habitacion". The data rows are:

#	idHabitacion	habitacion
1	1	Sala de examenes 1
2	2	Sala de examenes 2
3	3	Sala de examenes 3
4	4	Sala de examenes 4
5	5	Sala de imagenes 1
6	6	Sala de procedimientos 1

Imagen 10.3. SELECT a la tabla HABITACION.

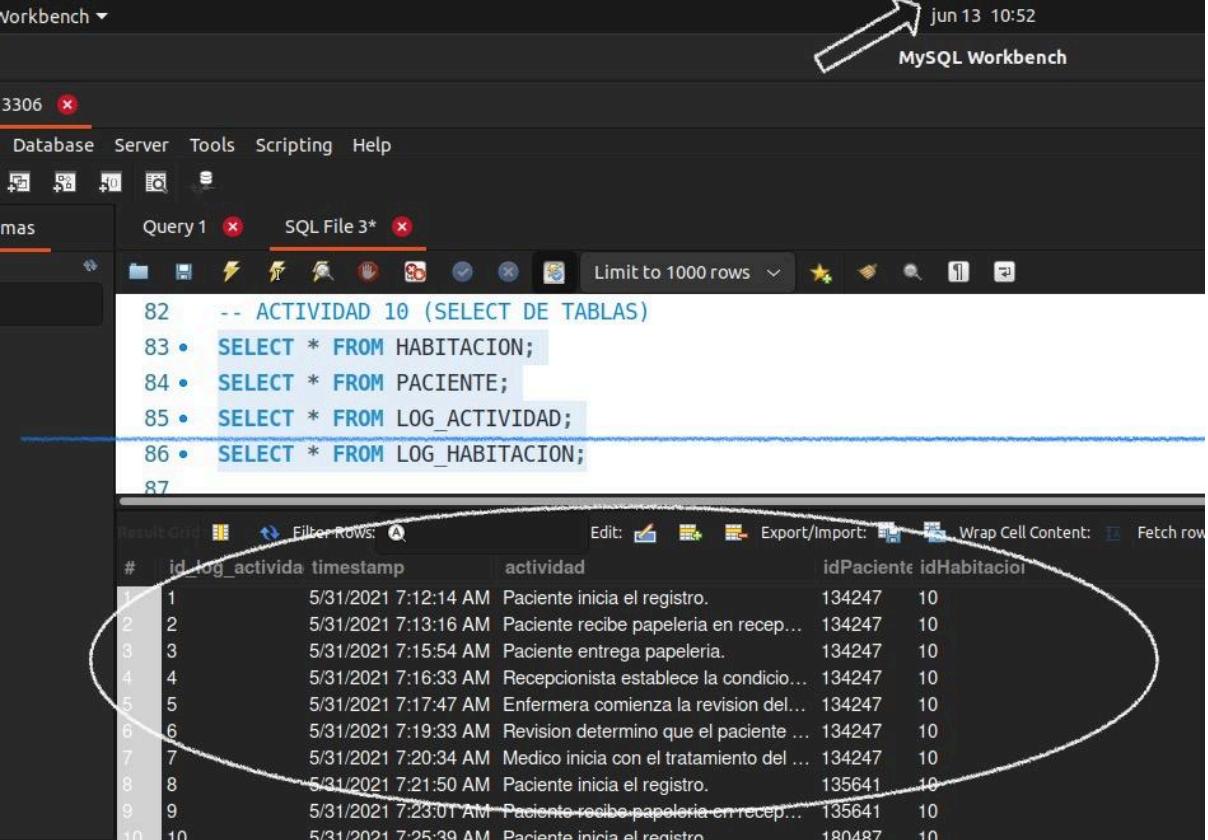
The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a menu bar with 'Database', 'Server', 'Tools', 'Scripting', and 'Help'. A status bar at the top right indicates 'jun 13 10:52' and 'MySQL Workbench'. The main area has two tabs: 'Query 1' and 'SQL File 3*'. The 'Query 1' tab contains the following SQL code:

```
82 -- ACTIVIDAD 10 (SELECT DE TABLAS)
83 • SELECT * FROM HABITACION;
84 • SELECT * FROM PACIENTE;
85 • SELECT * FROM LOG_ACTIVIDAD;
86 • SELECT * FROM LOG_HABITACION;
87
```

Below the code, a table is displayed with the following data:

#	idPaciente	edad	genero
1	100000	95	Otro
2	100001	40	Femenino
3	100002	42	Masculino
4	100003	8	Femenino
5	100004	88	Masculino
6	100005	23	Masculino
7	100006	60	Femenino
8	100007	49	Femenino
9	100008	44	Femenino
10	100009	55	Femenino

Imagen 10.4. SELECT a la tabla PACIENTE.



The screenshot shows the MySQL Workbench interface with a connection named 'mas' and port '3306'. The 'Query 1' tab is active, displaying the following SQL code:

```

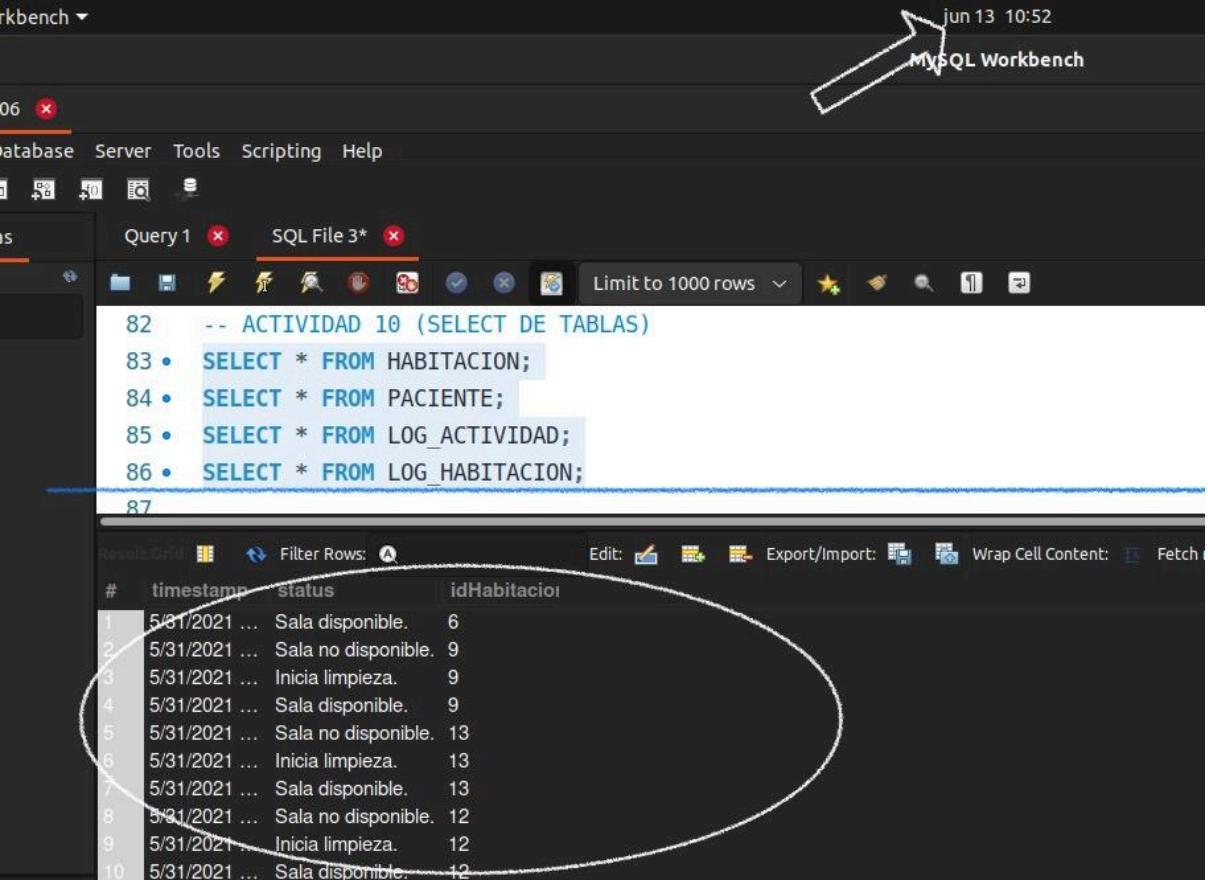
82 -- ACTIVIDAD 10 (SELECT DE TABLAS)
83 • SELECT * FROM HABITACION;
84 • SELECT * FROM PACIENTE;
85 • SELECT * FROM LOG_ACTIVIDAD;
86 • SELECT * FROM LOG_HABITACION;
87

```

The 'Result Grid' pane shows the data from the LOG_ACTIVIDAD table:

#	id_log_actividad	timestamp	actividad	idPaciente	idHabitacion
1	1	5/31/2021 7:12:14 AM	Paciente inicia el registro.	134247	10
2	2	5/31/2021 7:13:16 AM	Paciente recibe papeleria en recepcion...	134247	10
3	3	5/31/2021 7:15:54 AM	Paciente entrega papeleria.	134247	10
4	4	5/31/2021 7:16:33 AM	Recepcionista establece la condicion...	134247	10
5	5	5/31/2021 7:17:47 AM	Enfermera comienza la revision del...	134247	10
6	6	5/31/2021 7:19:33 AM	Revision determino que el paciente ...	134247	10
7	7	5/31/2021 7:20:34 AM	Medico inicia con el tratamiento del ...	134247	10
8	8	5/31/2021 7:21:50 AM	Paciente inicia el registro.	135641	10
9	9	5/31/2021 7:23:01 AM	Paciente recibe papeleria en recepcion...	135641	10
10	10	5/31/2021 7:25:39 AM	Paciente inicia el registro.	180487	10

Imagen 10.5. SELECT a la tabla LOG_ACTIVIDAD.



The screenshot shows the MySQL Workbench interface with a connection named 'mas' and port '3306'. The 'Query 1' tab is active, displaying the following SQL code:

```

82 -- ACTIVIDAD 10 (SELECT DE TABLAS)
83 • SELECT * FROM HABITACION;
84 • SELECT * FROM PACIENTE;
85 • SELECT * FROM LOG_ACTIVIDAD;
86 • SELECT * FROM LOG_HABITACION;
87

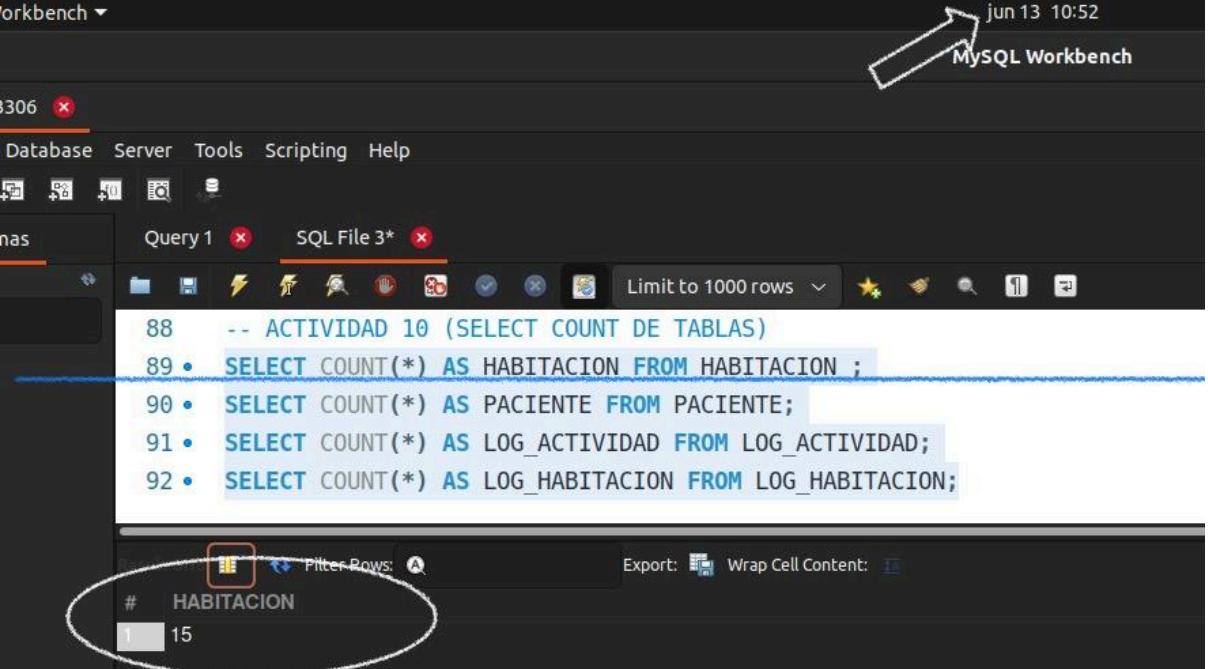
```

The 'Result Grid' pane shows the data from the LOG_HABITACION table:

#	timestamp	status	idHabitacion
1	5/31/2021 ...	Sala disponible.	6
2	5/31/2021 ...	Sala no disponible.	9
3	5/31/2021 ...	Inicia limpieza.	9
4	5/31/2021 ...	Sala disponible.	9
5	5/31/2021 ...	Sala no disponible.	13
6	5/31/2021 ...	Inicia limpieza.	13
7	5/31/2021 ...	Sala disponible.	13
8	5/31/2021 ...	Sala no disponible.	12
9	5/31/2021 ...	Inicia limpieza.	12
10	5/31/2021 ...	Sala disponible.	12

Imagen 10.6. SELECT a la tabla LOG_HABITACION.

Además de esto, como medida de seguridad ejecutamos el comando SELECT COUNT(*) sobre cada una de nuestras tablas.



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

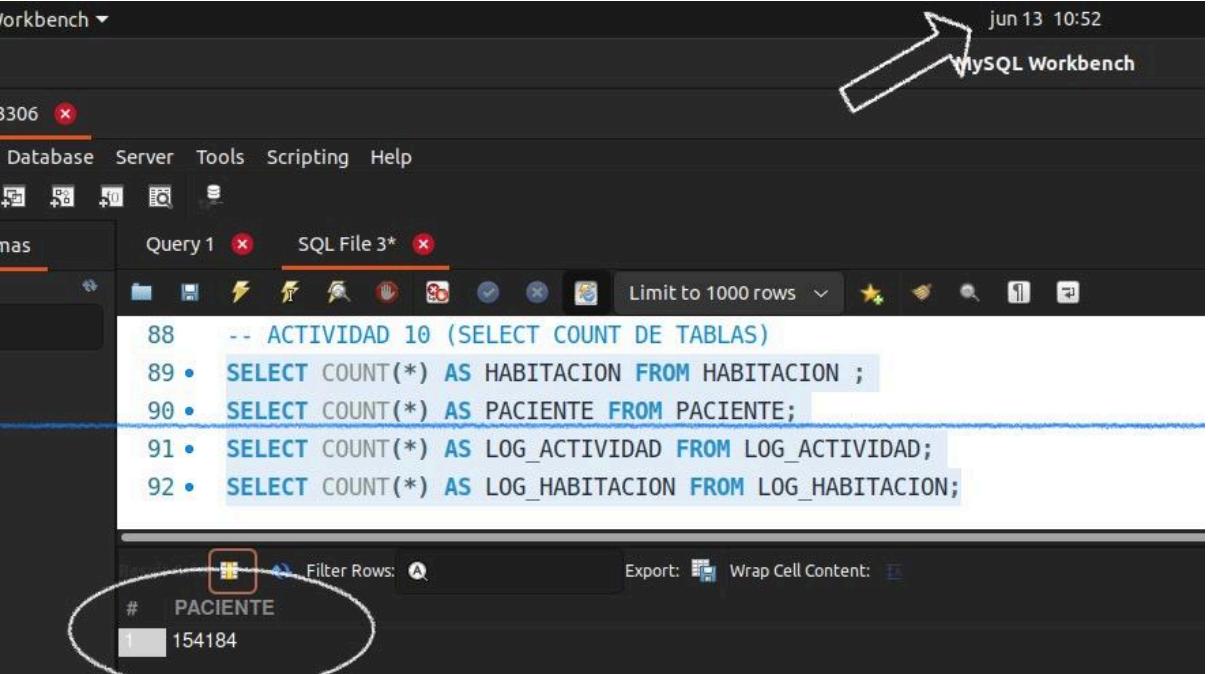
88 -- ACTIVIDAD 10 (SELECT COUNT DE TABLAS)
89 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
90 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE;
91 • SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD;
92 • SELECT COUNT(*) AS LOG_HABITACION FROM LOG_HABITACION;

```

The result set for the first query (HABITACION) is displayed below:

#	HABITACION
1	15

Imagen 10.7. SELECT COUNT(*) a la tabla HABITACION.



The screenshot shows the MySQL Workbench interface with a query editor containing the same SQL code as in Image 10.7:

```

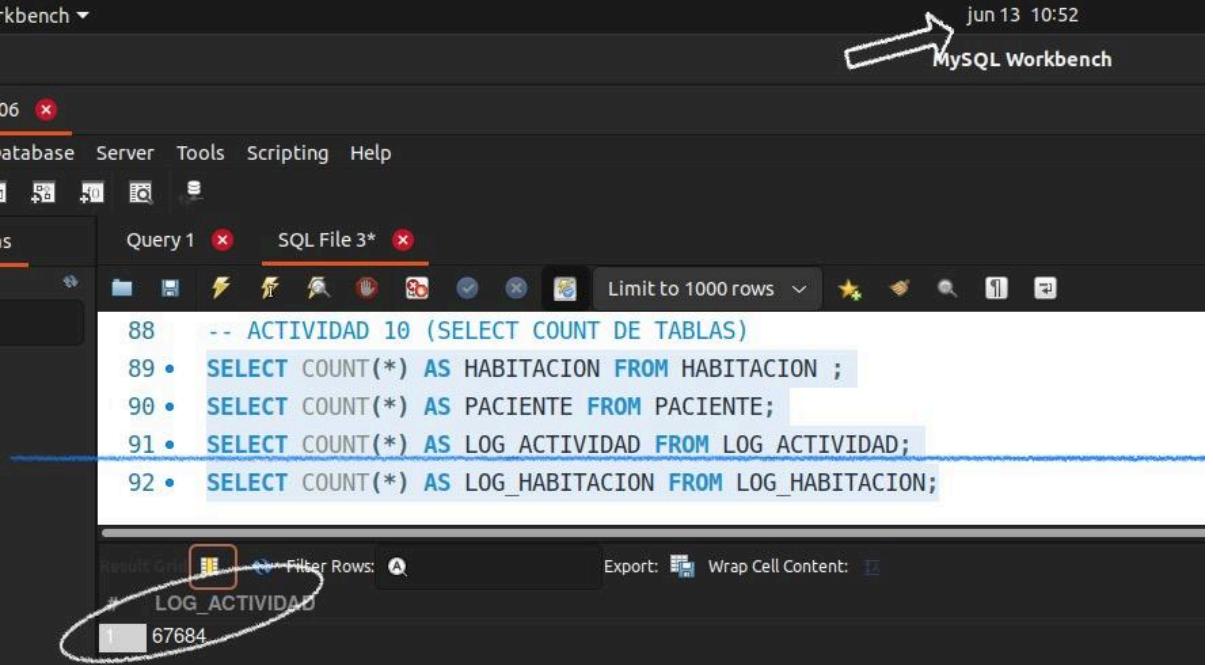
88 -- ACTIVIDAD 10 (SELECT COUNT DE TABLAS)
89 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
90 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE;
91 • SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD;
92 • SELECT COUNT(*) AS LOG_HABITACION FROM LOG_HABITACION;

```

The result set for the second query (PACIENTE) is displayed below:

#	PACIENTE
1	154184

Imagen 10.8. SELECT COUNT(*) a la tabla PACIENTE.



The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```

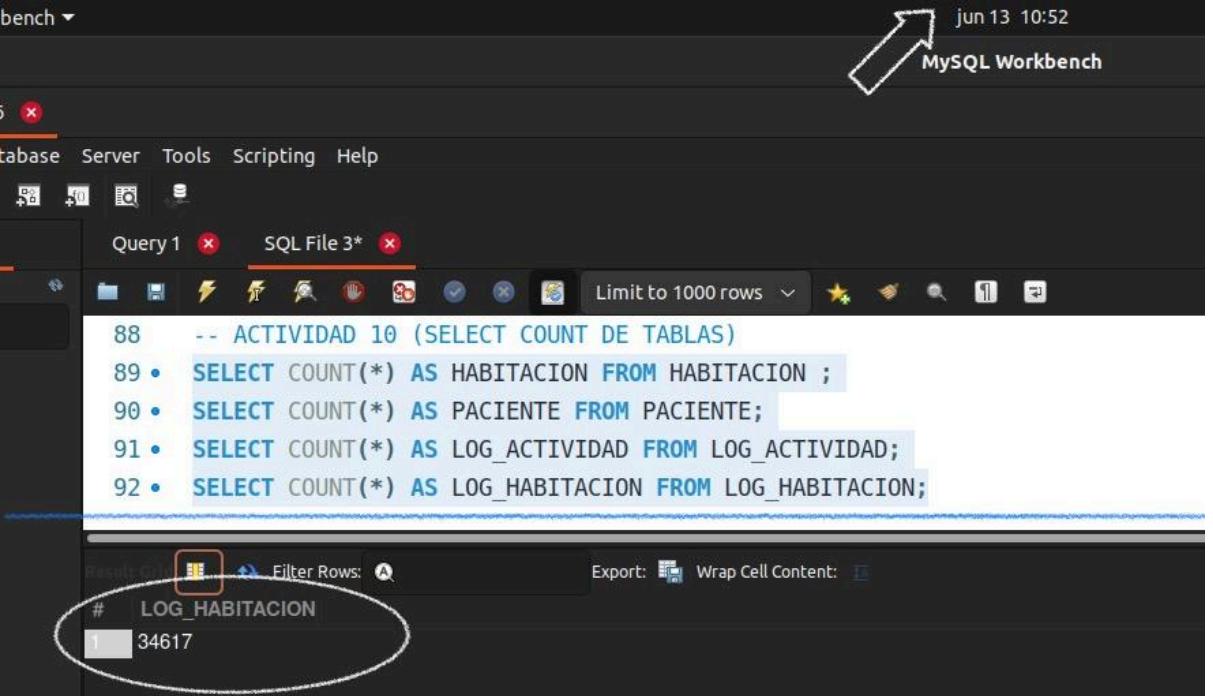
88 -- ACTIVIDAD 10 (SELECT COUNT DE TABLAS)
89 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
90 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE;
91 • SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD;
92 • SELECT COUNT(*) AS LOG_HABITACION FROM LOG_HABITACION;

```

The results pane displays the count for the LOG_ACTIVIDAD table:

LOG_ACTIVIDAD
1 67684

Imagen 10.9. SELECT COUNT(*) a la tabla LOG_ACTIVIDAD.



The screenshot shows the MySQL Workbench interface with a query editor containing the same SQL code as in Image 10.9:

```

88 -- ACTIVIDAD 10 (SELECT COUNT DE TABLAS)
89 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
90 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE;
91 • SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD;
92 • SELECT COUNT(*) AS LOG_HABITACION FROM LOG_HABITACION;

```

The results pane displays the count for the LOG_HABITACION table:

LOG_HABITACION
1 34617

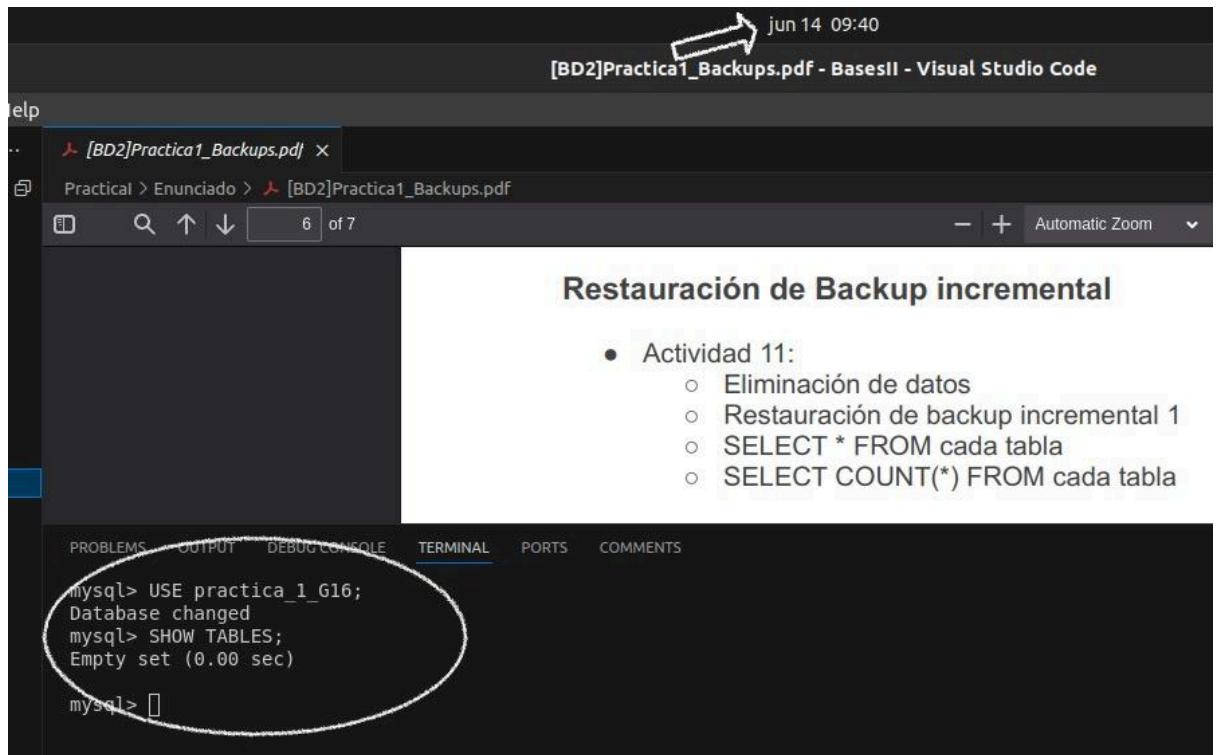
Imagen 10.10. SELECT COUNT(*) a la tabla LOG_HABITACION.

Restauración de backup incremental (Actividades 11-15)

Actividad 11

Para la primera actividad en la restauración de backups incrementales. Primero procedemos a eliminar los datos de nuestra base de datos. Esto al igual que en las actividades

anteriores, lo hicimos haciendo uso del comando `DROP TABLES` en nuestra base de datos. Dando como resultado lo siguiente.



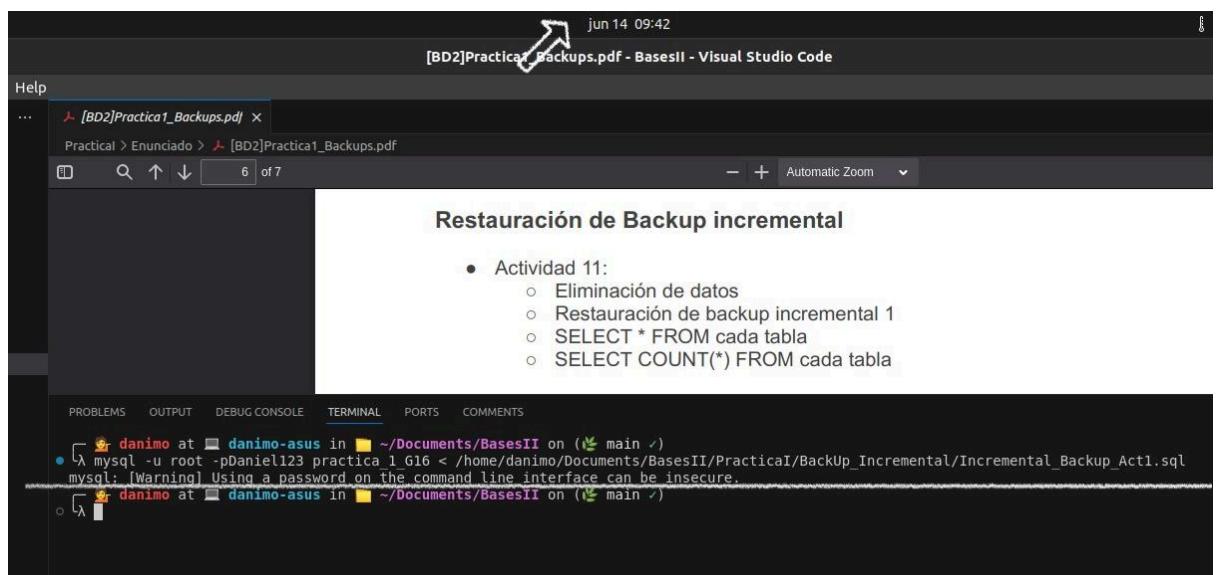
The screenshot shows a Visual Studio Code interface. At the top, there is a PDF viewer showing a document titled "Restauración de Backup incremental". Below it, the terminal tab is active, displaying the following MySQL session:

```
mysql> USE practica_1_G16;
Database changed
mysql> SHOW TABLES;
Empty set (0.00 sec)
mysql>
```

A white oval highlights the first two lines of the MySQL session.

Imagen 11.1. Eliminación de tablas en la base de datos.

Luego de haber confirmado que no existe información en nuestra base de datos debemos cargar los datos almacenados en el backup incremental 1 que se obtuvo en la Actividad 1.

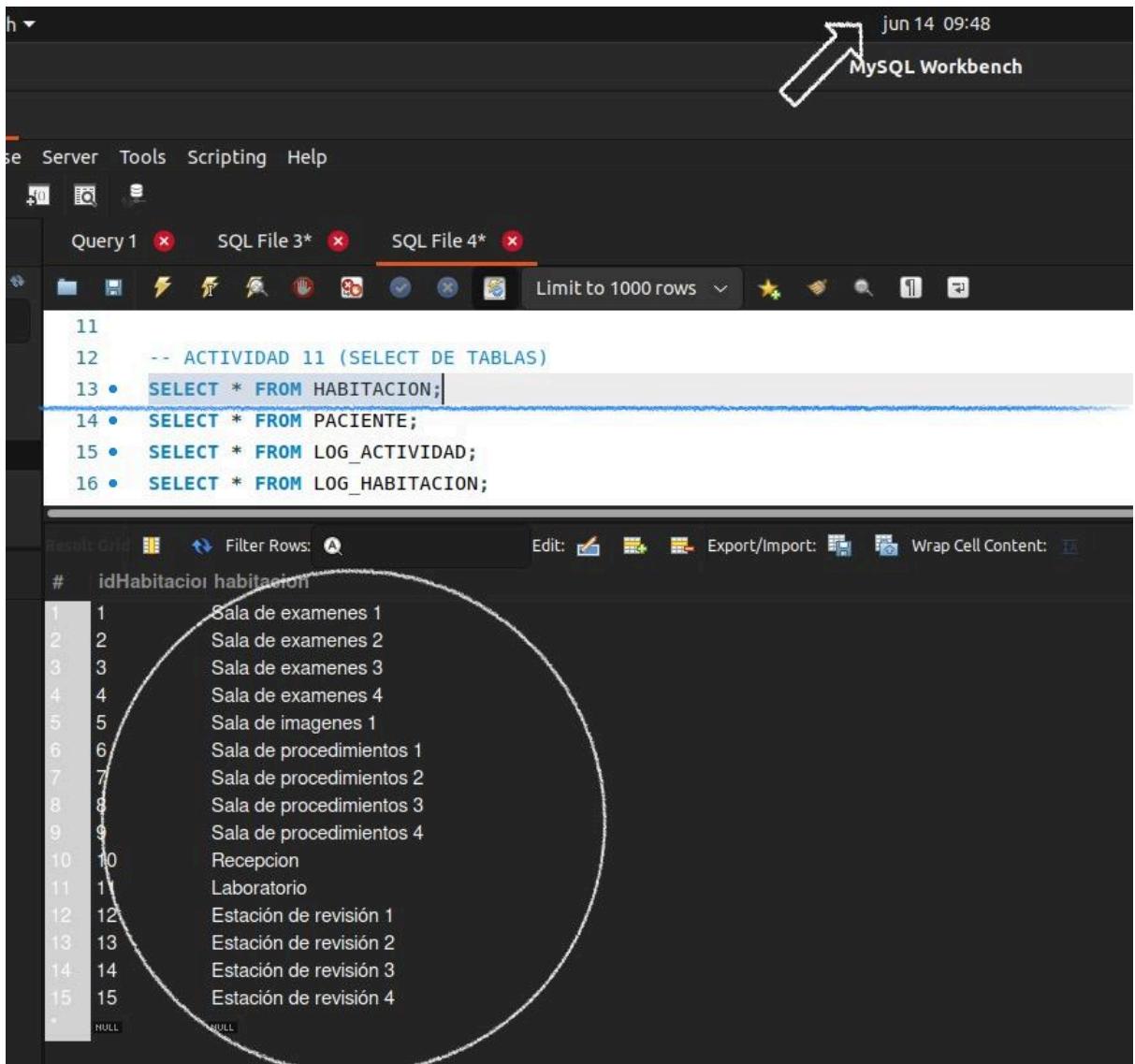


The screenshot shows a Visual Studio Code interface. At the top, there is a PDF viewer showing a document titled "Restauración de Backup incremental". Below it, the terminal tab is active, displaying the following MySQL session:

```
mysql> \! mysql -u root -pDaniel123 practica_1_G16 < /home/danimo/Documents/BasesII/PracticaI/BackUp_Incremental/Incremental_Backup_Act1.sql
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql>
```

Imagen 11.2. Ejecución de comando para restaurar nuestro backup incremental 1.

Una vez realizado el backup podemos verificar que el backup se haya realizado ejecutando el comando SELECT en la tabla HABITACION.



The screenshot shows the MySQL Workbench interface. At the top right, there is a timestamp 'Jun 14 09:48' and the title 'MySQL Workbench'. Below the title, there is a hand-drawn arrow pointing towards the title bar. The main window contains a SQL editor with four tabs: 'Query 1', 'SQL File 3*', 'SQL File 4*', and 'SQL File 4*' (the last one is highlighted). The SQL code in the editor is:

```

11
12 -- ACTIVIDAD 11 (SELECT DE TABLAS)
13 • SELECT * FROM HABITACION;
14 • SELECT * FROM PACIENTE;
15 • SELECT * FROM LOG_ACTIVIDAD;
16 • SELECT * FROM LOG_HABITACION;

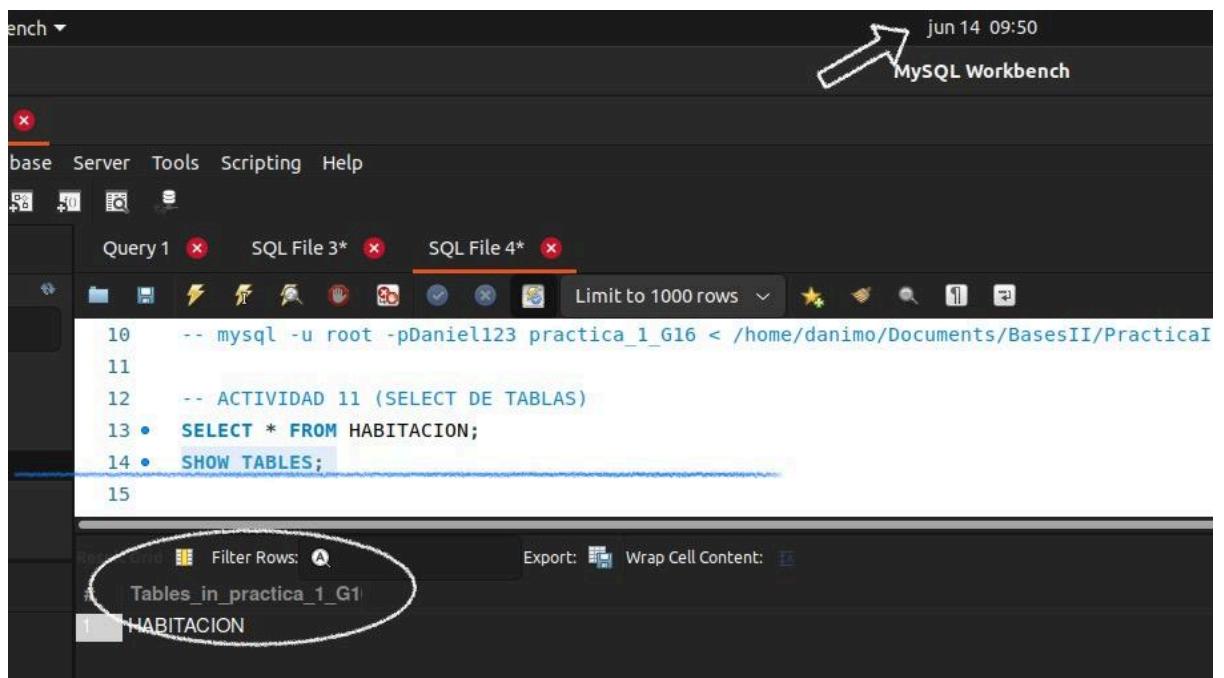
```

Below the editor is a 'Result Grid' section. It has a header row with columns '#', 'idHabitacion' (with a tooltip 'habitacion'), and 'habitacion'. The data grid contains 15 rows of data:

#	idHabitacion	habitacion
1	1	Sala de examenes 1
2	2	Sala de examenes 2
3	3	Sala de examenes 3
4	4	Sala de examenes 4
5	5	Sala de imagenes 1
6	6	Sala de procedimientos 1
7	7	Sala de procedimientos 2
8	8	Sala de procedimientos 3
9	9	Sala de procedimientos 4
10	10	Recepcion
11	11	Laboratorio
12	12	Estación de revisión 1
13	13	Estación de revisión 2
14	14	Estación de revisión 3
15	15	Estación de revisión 4
*	HULL	NULL

Imagen 11.3. SELECT a la tabla HABITACION.

Es necesario aclarar que a diferencia de las actividades de restauración de full backup, en este caso no se puede hacer un SELECT a las otras tablas debido a que en este caso se eliminaron y el backup incremental únicamente posee la creación de la tabla a la que se le hizo backup. Esto se puede confirmar al momento de ejecutar el comando SHOW TABLES en nuestra base de datos.



```

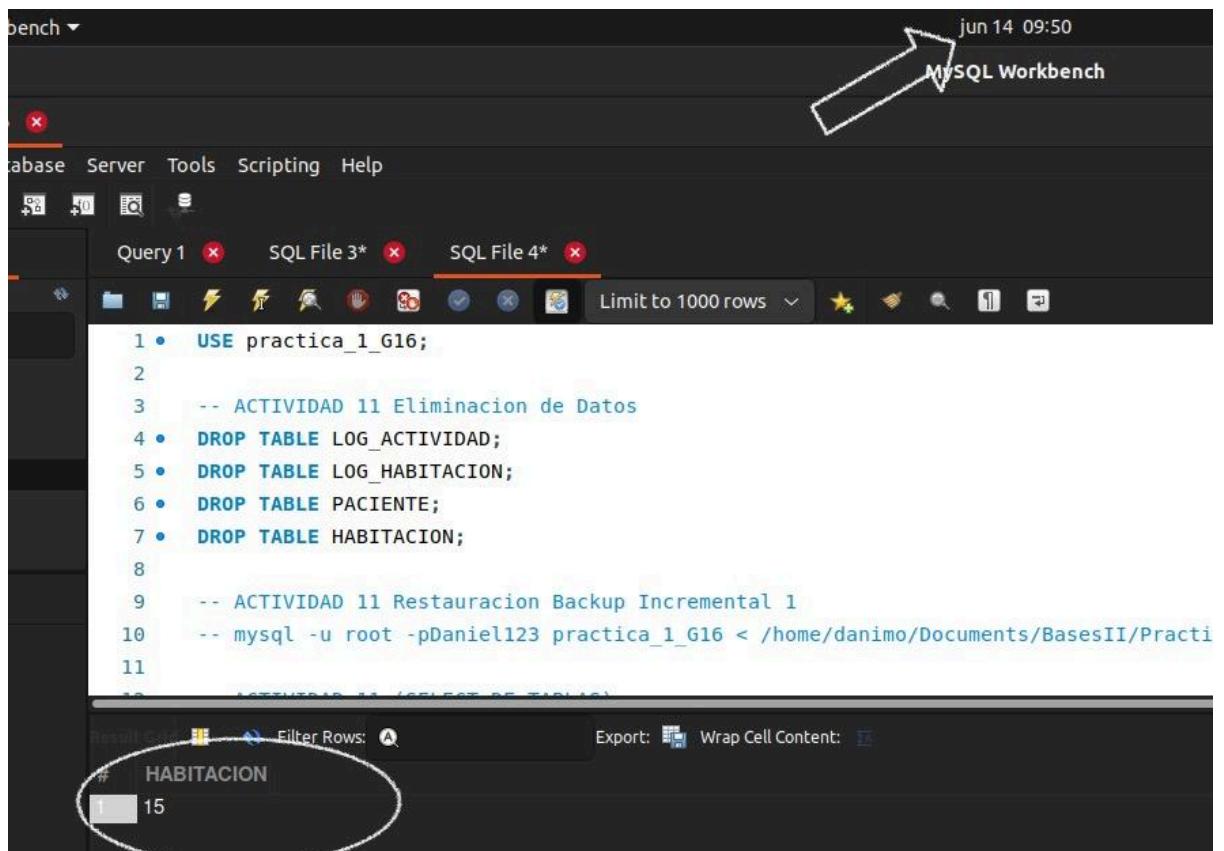
10 -- mysql -u root -pDaniel123 practica_1_G16 < /home/danimo/Documents/BasesII/PracticaI
11
12 -- ACTIVIDAD 11 (SELECT DE TABLAS)
13 • SELECT * FROM HABITACION;
14 • SHOW TABLES;
15

```

#	Tables_in_practica_1_G1
1	HABITACION

Imagen 11.4. SHOW TABLES en nuestra base de datos.

También podemos hacer uso del comando `SELECT COUNT(*)` para verificar que se hayan cargado los datos.



```

1 • USE practica_1_G16;
2
3 -- ACTIVIDAD 11 Eliminacion de Datos
4 • DROP TABLE LOG_ACTIVIDAD;
5 • DROP TABLE LOG_HABITACION;
6 • DROP TABLE PACIENTE;
7 • DROP TABLE HABITACION;
8
9 -- ACTIVIDAD 11 Restauracion Backup Incremental 1
10 -- mysql -u root -pDaniel123 practica_1_G16 < /home/danimo/Documents/BasesII/PracticaI
11
12 -- ACTIVIDAD 11 (SELECT DE TABLAS)

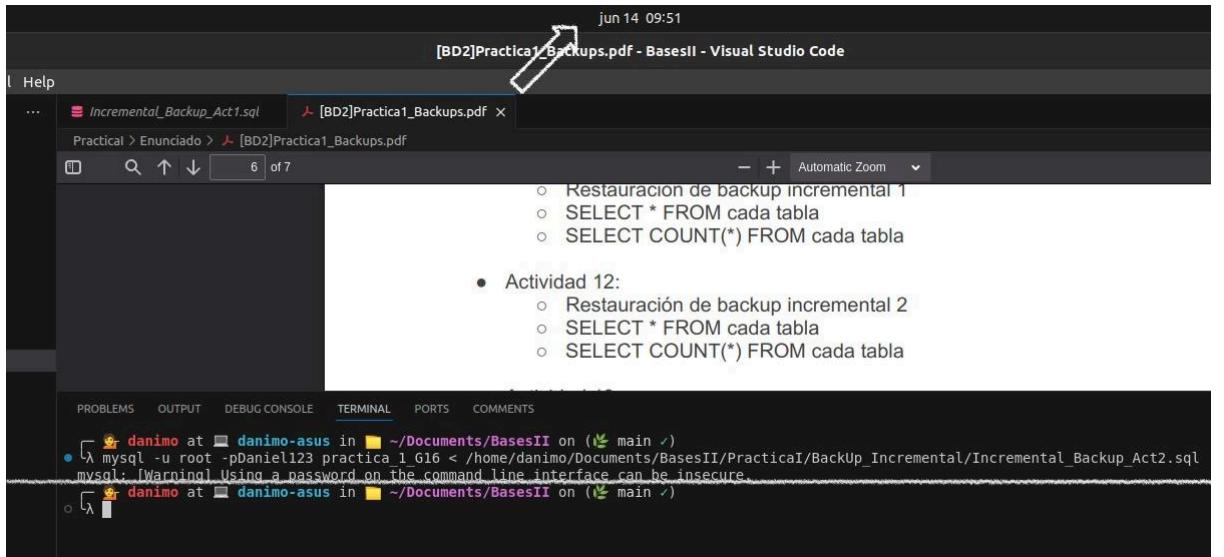
```

#	
1	15

Imagen 11.5. `SELECT COUNT(*)` a la tabla HABITACION.

Actividad 12

Para la segunda actividad en la restauración de backups incrementales. Tomando en cuenta que ya se debió de haber cargado el backup incremental anterior, ejecutamos el backup incremental 2 obtenido en la Actividad 2.



The screenshot shows a Visual Studio Code interface. At the top, there's a status bar with the date 'jun 14 09:51' and a tab labeled '[BD2]Practica1_Backups.pdf - BasesII - Visual Studio Code'. Below the status bar is a navigation bar with 'Help', 'Incremental_Backup_Act1.sql', '[BD2]Practica1_Backups.pdf', and other options like 'Practical > Enunciado'. The main area has tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', 'PORTS', and 'COMMENTS'. The 'TERMINAL' tab is active, showing a MySQL session:

```
danimo at danimo-asus in ~/Documents/BasesII on (main) 
mysql -u root -pDaniel123 practica1 G16 < /home/danimo/Documents/BasesII/PracticaI/BackUp_Incremental/Incremental_Backup_Act2.sql
mysql: Warning! Using a password on the command line interface can be insecure.

```

Imagen 12.1. Ejecución de comando para restaurar nuestro backup incremental 2.

Una vez realizado el backup podemos verificar que el backup se haya realizado ejecutando el comando SELECT en las tablas HABITACION y PACIENTE.

The screenshot shows the MySQL Workbench interface. In the top right corner, there is a timestamp 'jun 14 09:53' and the title 'MySQL Workbench'. The main window has a dark theme. At the top, there is a menu bar with 'base', 'Server', 'Tools', 'Scripting', and 'Help'. Below the menu is a toolbar with various icons. The central area contains three tabs: 'Query 1', 'SQL File 3*', and 'SQL File 4*'. The 'Query 1' tab is active and displays the following SQL code:

```
23 -- ACTIVIDAD 11 (SELECT DE TABLAS)
24 • SELECT * FROM HABITACION;
25 • SELECT * FROM PACIENTE;
26 • SHOW TABLES;
27
```

Below the code, the results are displayed in a table titled 'Result Grid'. The table has two columns: '# idHabitacion' and 'habitacion'. The data is as follows:

#	idHabitacion	habitacion
1	1	Sala de exámenes 1
2	2	Sala de exámenes 2
3	3	Sala de exámenes 3
4	4	Sala de exámenes 4
5	5	Sala de imágenes 1
6	6	Sala de procedimientos 1
7	7	Sala de procedimientos 2
8	8	Sala de procedimientos 3
9	9	Sala de procedimientos 4
10	10	Recepción
11	11	Laboratorio
12	12	Estación de revisión 1
13	13	Estación de revisión 2
14	14	Estación de revisión 3
15	15	Estación de revisión 4
	NULL	NULL

Imagen 12.2. SELECT a la tabla HABITACION.

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a menu bar with 'Database', 'Server', 'Tools', 'Scripting', and 'Help'. There are tabs for 'Query 1', 'SQL File 3*', and 'SQL File 4*'. A status bar at the top right shows 'jun 14 09:53' and the 'MySQL Workbench' logo. The main area contains a SQL editor with the following code:

```
23 -- ACTIVIDAD 11 (SELECT DE TABLAS)
24 • SELECT * FROM HABITACION;
25 • SELECT * FROM PACIENTE;
26 • SHOW TABLES;
27
```

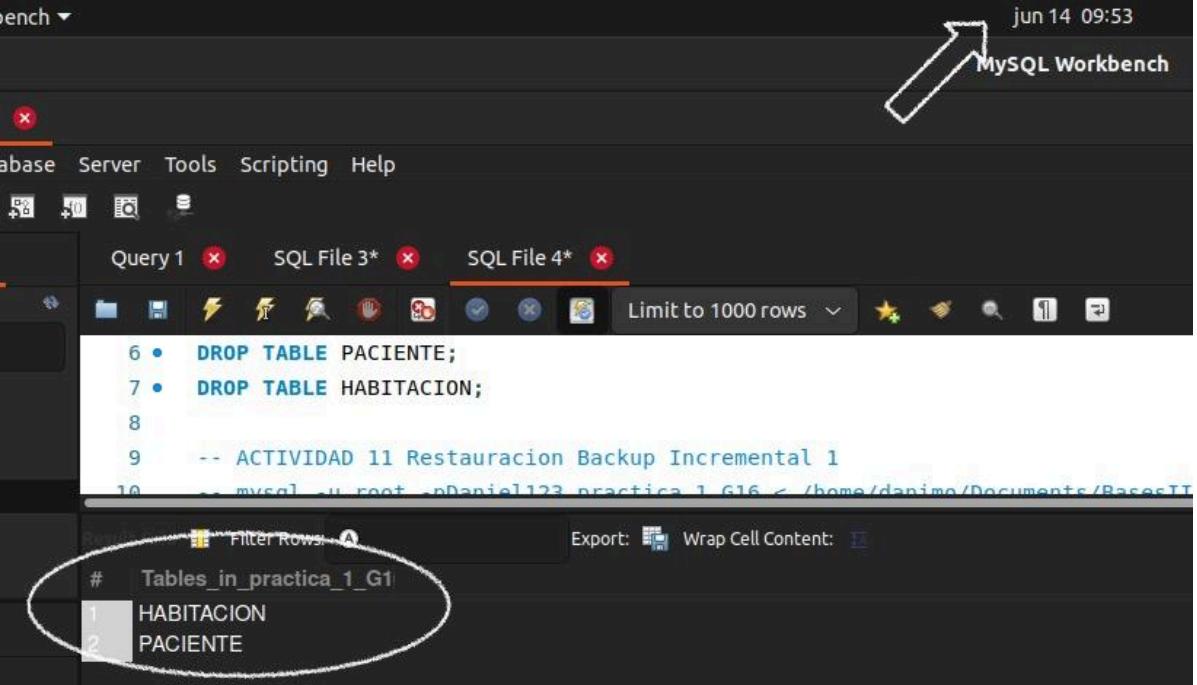
Below the editor is a 'Result Grid' table with the following data:

#	idPaciente	edad	genero
17	100016	39	Masculino
18	100017	55	Femenino
19	100018	61	Masculino
20	100019	58	Masculino
21	100020	6	Masculino
22	100021	26	Femenino
23	100022	2	Masculino
24	100023	6	Femenino
25	100024	97	Masculino
26	100025	67	Femenino
27	100026	57	Femenino
28	100027	54	Femenino
29	100028	32	Femenino
30	100029	53	Masculino
31			

A large white circle highlights the row for idPaciente 100026. The row for idPaciente 100026 is also highlighted with a red background.

Imagen 12.3. SELECT a la tabla PACIENTE.

Si ejecutamos el comando SHOW TABLES podremos notar que se ha agregado solamente la tabla PACIENTE a nuestra base de datos.



The screenshot shows the MySQL Workbench interface. In the top right corner, there is a timestamp 'jun 14 09:53' and a yellow arrow pointing to the title bar which says 'MySQL Workbench'. The main window has a menu bar with 'Database', 'Server', 'Tools', 'Scripting', and 'Help'. Below the menu is a toolbar with various icons. There are three tabs at the top: 'Query 1', 'SQL File 3*', and 'SQL File 4*'. The 'Query 1' tab is active and contains the following SQL code:

```

6 • DROP TABLE PACIENTE;
7 • DROP TABLE HABITACION;
8
9 -- ACTIVIDAD 11 Restauracion Backup Incremental 1
10 -- mysql -u root -pDaniel123 practica_1_G16 < /home/danimo/Documents/BasesII/Practica_1_G16.sql

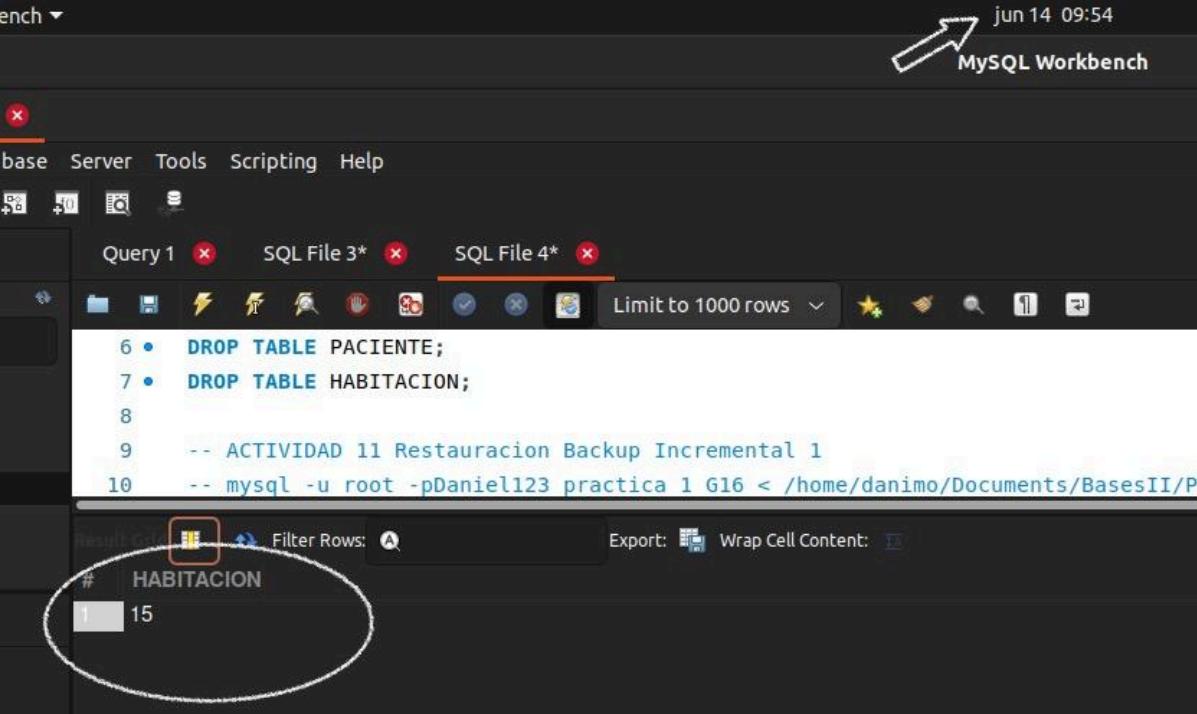
```

Below the code, the results of a 'SHOW TABLES' query are displayed in a table:

#	Tables_in_practica_1_G1
1	HABITACION
2	PACIENTE

Imagen 12.4. SHOW TABLES a nuestra base de datos.

También podemos hacer uso del comando SELECT COUNT(*) para verificar que se hayan cargado los datos.



The screenshot shows the MySQL Workbench interface. In the top right corner, there is a timestamp 'jun 14 09:54' and a yellow arrow pointing to the title bar which says 'MySQL Workbench'. The main window has a menu bar with 'Database', 'Server', 'Tools', 'Scripting', and 'Help'. Below the menu is a toolbar with various icons. There are three tabs at the top: 'Query 1', 'SQL File 3*', and 'SQL File 4*'. The 'Query 1' tab is active and contains the following SQL code:

```

6 • DROP TABLE PACIENTE;
7 • DROP TABLE HABITACION;
8
9 -- ACTIVIDAD 11 Restauracion Backup Incremental 1
10 -- mysql -u root -pDaniel123 practica_1_G16 < /home/danimo/Documents/BasesII/Practica_1_G16.sql

```

Below the code, the results of a 'SELECT COUNT(*)' query on the 'HABITACION' table are displayed in a table:

#	HABITACION
1	15

Imagen 12.5. SELECT COUNT(*) a la tabla HABITACION.

The screenshot shows the MySQL Workbench interface. In the top right corner, there is a red arrow pointing to the status bar which displays "jun 14 09:55". The main window has a menu bar with "base", "Server", "Tools", "Scripting", and "Help". Below the menu is a toolbar with various icons. There are four tabs at the top: "Query 1", "SQL File 3*", "SQL File 4*", and "SQL File 4*". The "SQL File 4*" tab is active, showing the following SQL code:

```

28 -- ACTIVIDAD 12 (SELECT COUNT DE TABLAS)
29 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
30 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE ;
31 • SHOW TABLES;
32

```

Below the code, the results are displayed in a table titled "PACIENTE". The table has one row with the value 154184.

Imagen 12.6. SELECT COUNT(*) a la tabla PACIENTE.

Actividad 13

Para la segunda actividad en la restauración de backups incrementales. Tomando en cuenta que ya se debió de haber cargado el backup incremental anterior, ejecutamos el backup incremental 3 obtenido en la Actividad 3.

The screenshot shows the Visual Studio Code interface. At the top, there is a red arrow pointing to the title bar which displays "[BD2]Practica1_Backups.pdf - BasesII - Visual Studio Code". The left sidebar shows a tree view with nodes like "Restauracion de backup incremental 2", "SELECT * FROM cada tabla", "SELECT COUNT(*) FROM cada tabla", and "Actividad 13:" followed by three sub-nodes: "Restauracion de backup incremental 3", "SELECT * FROM cada tabla", and "SELECT COUNT(*) FROM cada tabla". The bottom part of the screen shows a terminal window with the following command and output:

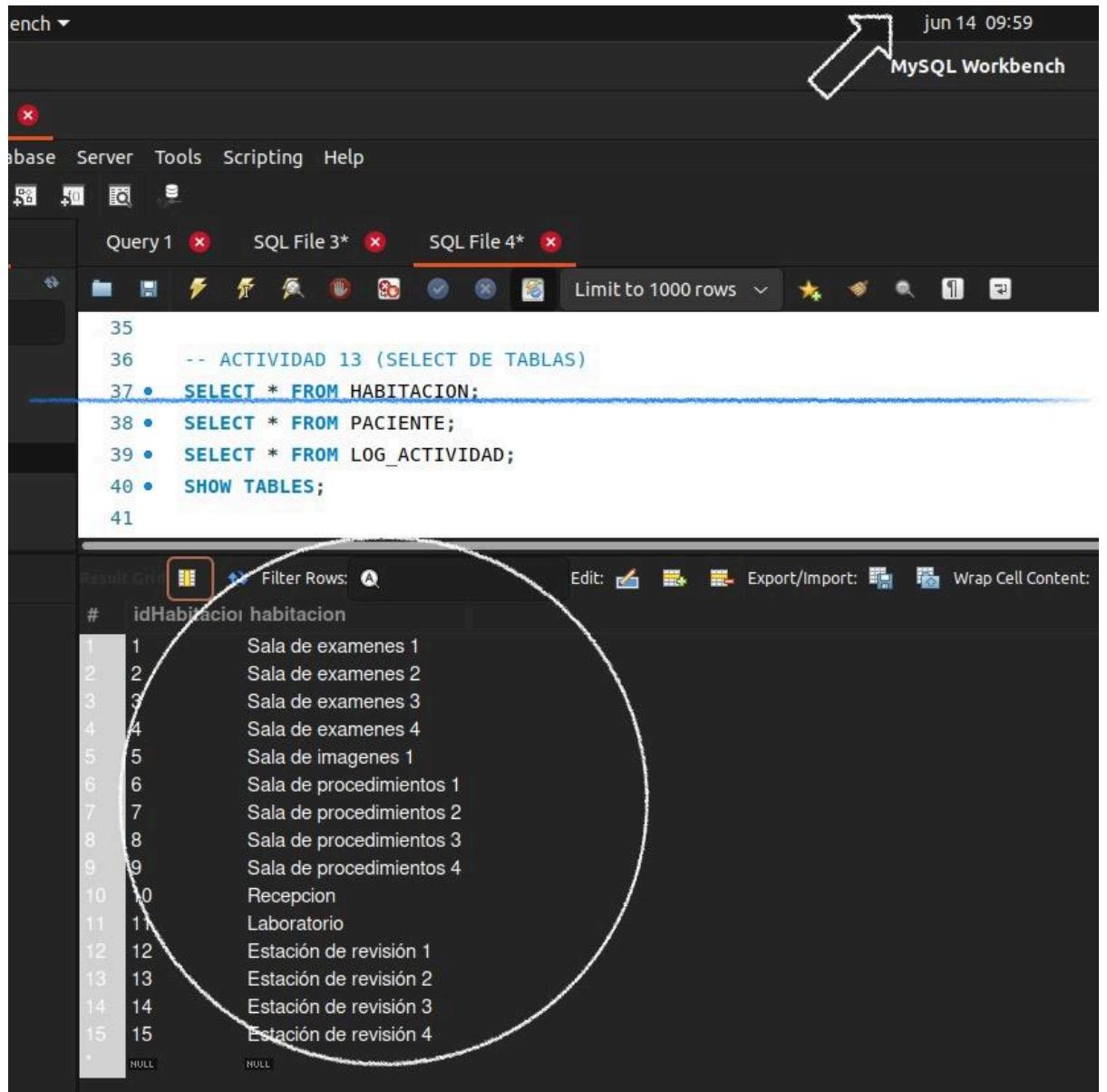
```

mysql -u root -pDaniel123 practica_1 G16 < /home/danimo/Documents/BasesII/PracticaI/BackUp_Incremental/Incremental_Backup_Act3.sql
mysql: [Warning] Using a password on the command line interface can be insecure.

```

Imagen 13.1. Ejecución de comando para restaurar nuestro backup incremental 3.

Una vez realizado el backup podemos verificar que el backup se haya realizado ejecutando el comando SELECT en las tablas HABITACION, PACIENTE y LOG_ACTIVIDAD.



The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query editor contains the following SQL code:

```
35
36 -- ACTIVIDAD 13 (SELECT DE TABLAS)
37 • SELECT * FROM HABITACION;
38 • SELECT * FROM PACIENTE;
39 • SELECT * FROM LOG_ACTIVIDAD;
40 • SHOW TABLES;
41
```

The results grid displays the data from the HABITACION table, with columns labeled '#', 'idHabitacion' (which is the primary key), and 'habitacion'. The data is as follows:

#	idHabitacion	habitacion
1	1	Sala de exámenes 1
2	2	Sala de exámenes 2
3	3	Sala de exámenes 3
4	4	Sala de exámenes 4
5	5	Sala de imágenes 1
6	6	Sala de procedimientos 1
7	7	Sala de procedimientos 2
8	8	Sala de procedimientos 3
9	9	Sala de procedimientos 4
10	10	Recepción
11	11	Laboratorio
12	12	Estación de revisión 1
13	13	Estación de revisión 2
14	14	Estación de revisión 3
15	15	Estación de revisión 4
*	HULL	HULL

Imagen 13.2. SELECT a la tabla HABITACION.

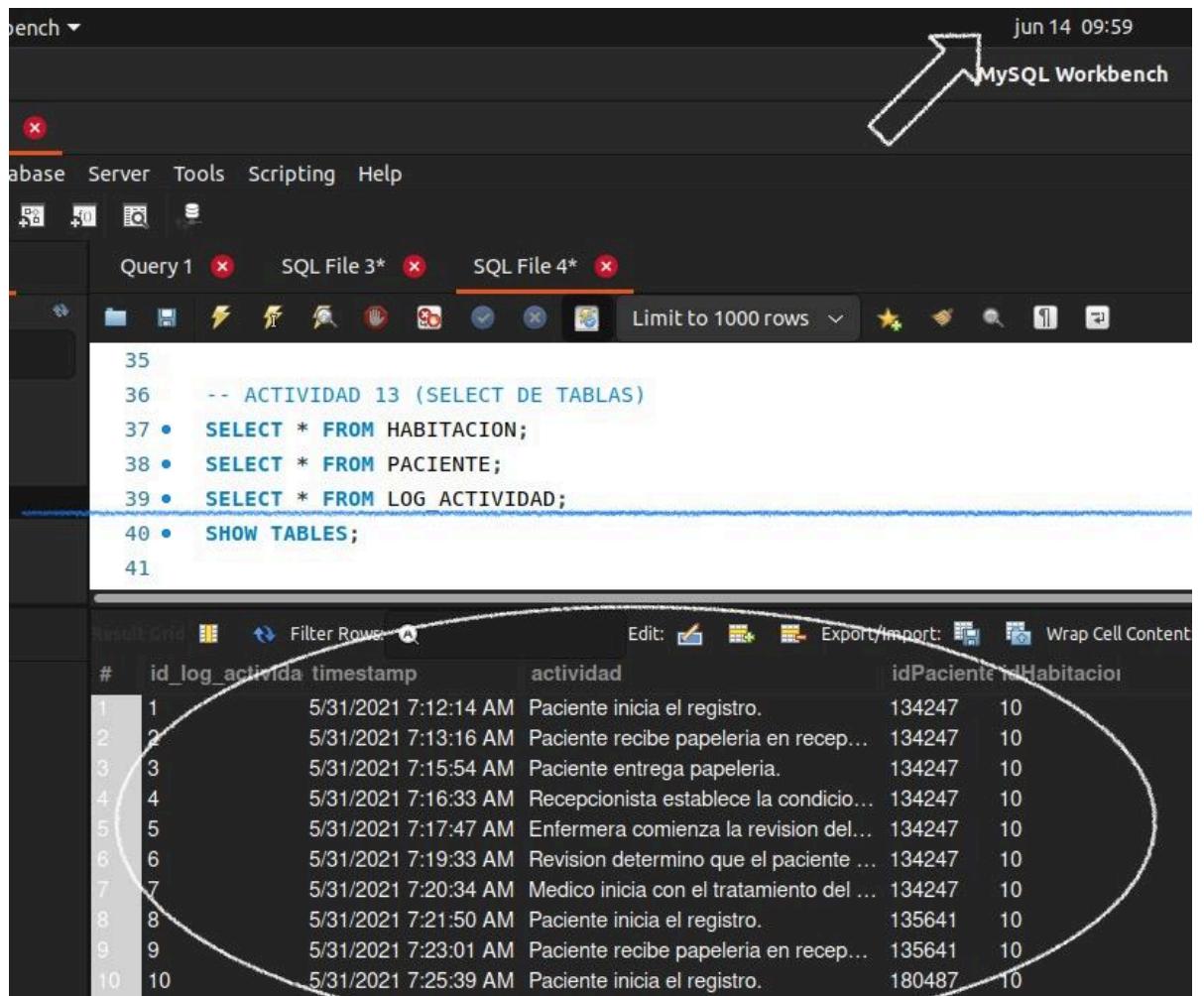
The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it, a menu bar with 'Database', 'Server', 'Tools', 'Scripting', and 'Help'. There are three tabs open: 'Query 1', 'SQL File 3*', and 'SQL File 4*'. The 'SQL File 4*' tab is active, containing the following SQL code:

```
35
36 -- ACTIVIDAD 13 (SELECT DE TABLAS)
37 • SELECT * FROM HABITACION;
38 • SELECT * FROM PACIENTE;
39 • SELECT * FROM LOG_ACTIVIDAD;
40 • SHOW TABLES;
41
```

Below the code, the 'Result' tab is selected, showing the output of the 'SELECT * FROM PACIENTE;' query. The results are presented in a table with columns: '#', 'idPaciente', 'edad', and 'genero'. The data is as follows:

#	idPaciente	edad	genero
1	100000	95	Otro
2	100001	40	Femenino
3	100002	42	Masculino
4	100003	8	Femenino
5	100004	88	Masculino
6	100005	23	Masculino
7	100006	60	Femenino
8	100007	49	Femenino
9	100008	44	Femenino
10	100009	55	Femenino
11	100010	28	Femenino
12	100011	13	Femenino
13	100012	60	Masculino

Imagen 13.3. SELECT a la tabla PACIENTE.



The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it, the menu bar includes Database, Server, Tools, Scripting, and Help. There are four tabs open in the main area: Query 1, SQL File 3*, and SQL File 4*. A red arrow points from the text "Imagen 13.4. SELECT a la tabla LOG_ACTIVIDAD." to the results grid.

```

35
36      -- ACTIVIDAD 13 (SELECT DE TABLAS)
37 •  SELECT * FROM HABITACION;
38 •  SELECT * FROM PACIENTE;
39 •  SELECT * FROM LOG_ACTIVIDAD;
40 •  SHOW TABLES;
41

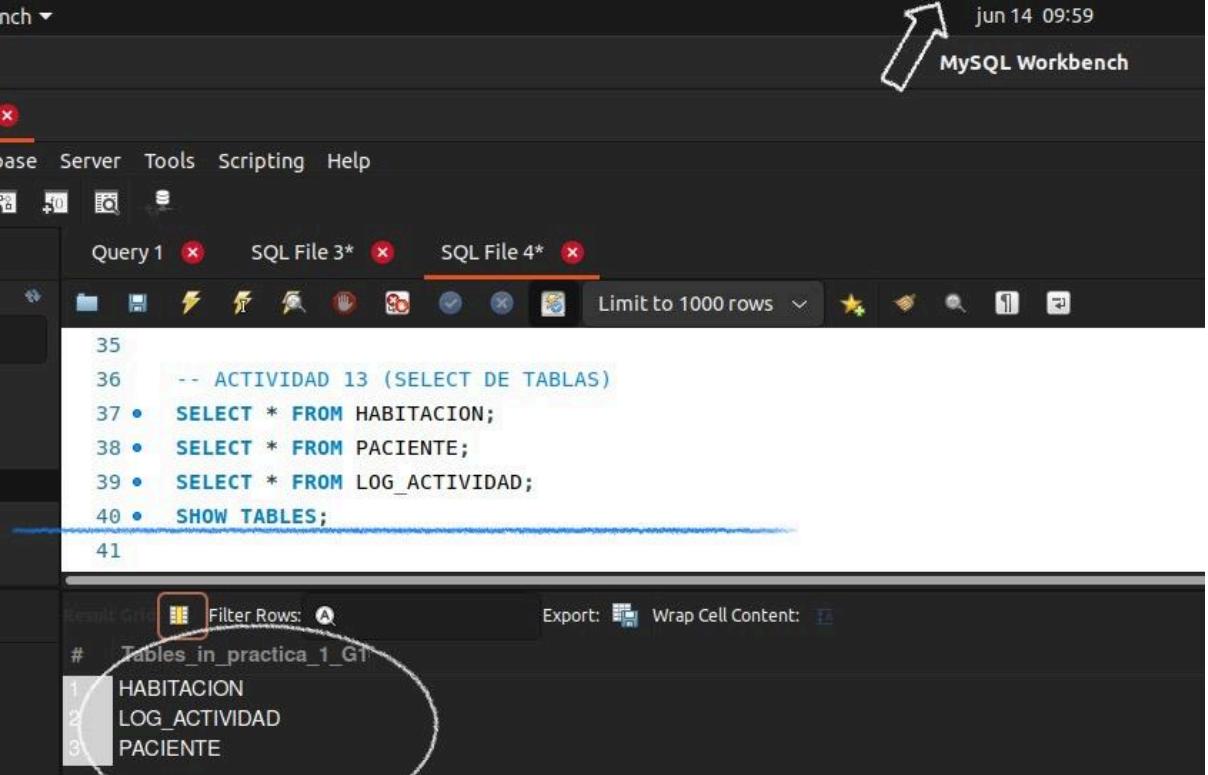
```

The results grid displays data from the LOG_ACTIVIDAD table:

#	id_log_actividad	timestamp	actividad	idPaciente	idHabitacion
1	1	5/31/2021 7:12:14 AM	Paciente inicia el registro.	134247	10
2	2	5/31/2021 7:13:16 AM	Paciente recibe papeleria en recepcion...	134247	10
3	3	5/31/2021 7:15:54 AM	Paciente entrega papeleria.	134247	10
4	4	5/31/2021 7:16:33 AM	Recepcionista establece la condicion...	134247	10
5	5	5/31/2021 7:17:47 AM	Enfermera comienza la revision del...	134247	10
6	6	5/31/2021 7:19:33 AM	Revision determino que el paciente ...	134247	10
7	7	5/31/2021 7:20:34 AM	Medico inicia con el tratamiento del ...	134247	10
8	8	5/31/2021 7:21:50 AM	Paciente inicia el registro.	135641	10
9	9	5/31/2021 7:23:01 AM	Paciente recibe papeleria en recepcion...	135641	10
10	10	5/31/2021 7:25:39 AM	Paciente inicia el registro.	180487	10

Imagen 13.4. SELECT a la tabla LOG_ACTIVIDAD.

Si ejecutamos el comando SHOW TABLES podremos notar que se ha agregado solamente la tabla LOG_ACTIVIDAD a nuestra base de datos.



The screenshot shows the MySQL Workbench interface. In the top right corner, it says "jun 14 09:59" and "MySQL Workbench". The menu bar includes "Base", "Server", "Tools", "Scripting", and "Help". Below the menu is a toolbar with various icons. There are three tabs open: "Query 1", "SQL File 3*", and "SQL File 4*". The "Query 1" tab contains the following SQL code:

```

35
36 -- ACTIVIDAD 13 (SELECT DE TABLAS)
37 • SELECT * FROM HABITACION;
38 • SELECT * FROM PACIENTE;
39 • SELECT * FROM LOG_ACTIVIDAD;
40 • SHOW TABLES;
41

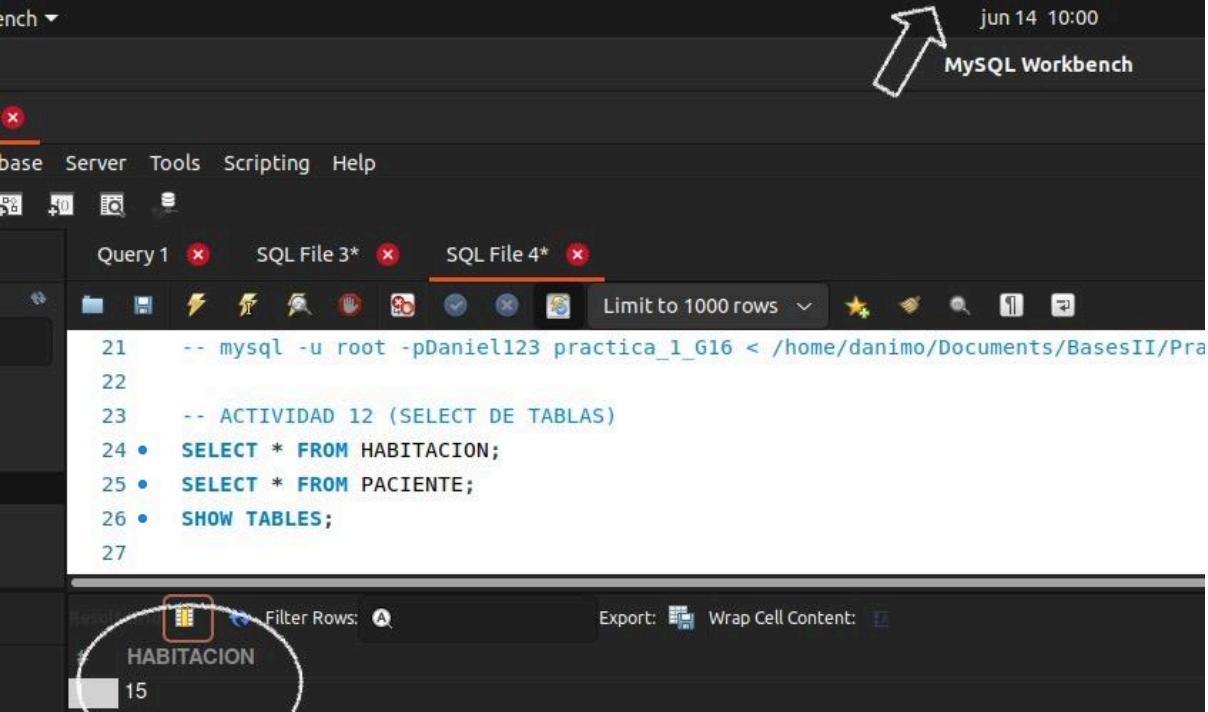
```

The results grid shows the output of the "SHOW TABLES" command:

#	Tables_in_practica_1_G1
1	HABITACION
2	LOG_ACTIVIDAD
3	PACIENTE

Imagen 13.5. SHOW TABLES a nuestra base de datos.

También podemos hacer uso del comando `SELECT COUNT(*)` para verificar que se hayan cargado los datos.



The screenshot shows the MySQL Workbench interface. In the top right corner, it says "jun 14 10:00" and "MySQL Workbench". The menu bar includes "Base", "Server", "Tools", "Scripting", and "Help". Below the menu is a toolbar with various icons. There are three tabs open: "Query 1", "SQL File 3*", and "SQL File 4*". The "Query 1" tab contains the following SQL code:

```

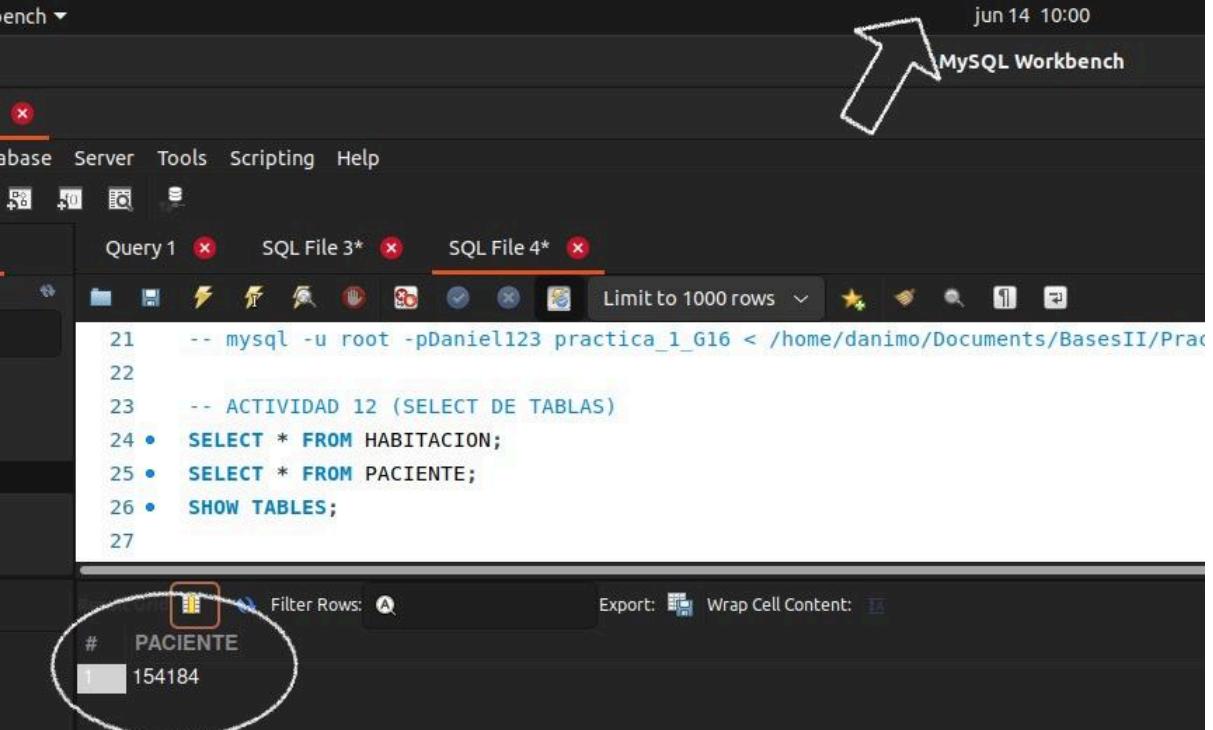
21 -- mysql -u root -pDaniel123 practica_1_G16 < /home/danimo/Documents/BasesII/Pra
22
23 -- ACTIVIDAD 12 (SELECT DE TABLAS)
24 • SELECT * FROM HABITACION;
25 • SELECT * FROM PACIENTE;
26 • SHOW TABLES;
27

```

The results grid shows the output of the `SELECT COUNT(*)` query on the "HABITACION" table:

	HABITACION
15	

Imagen 13.6. `SELECT COUNT(*)` a la tabla HABITACION.



The screenshot shows the MySQL Workbench interface. The title bar says "jun 14 10:00 MySQL Workbench". The menu bar includes Database, Server, Tools, Scripting, and Help. The toolbar has various icons for database management. Below the toolbar are three tabs: "Query 1", "SQL File 3*", and "SQL File 4*". The "Query 1" tab contains the following SQL code:

```

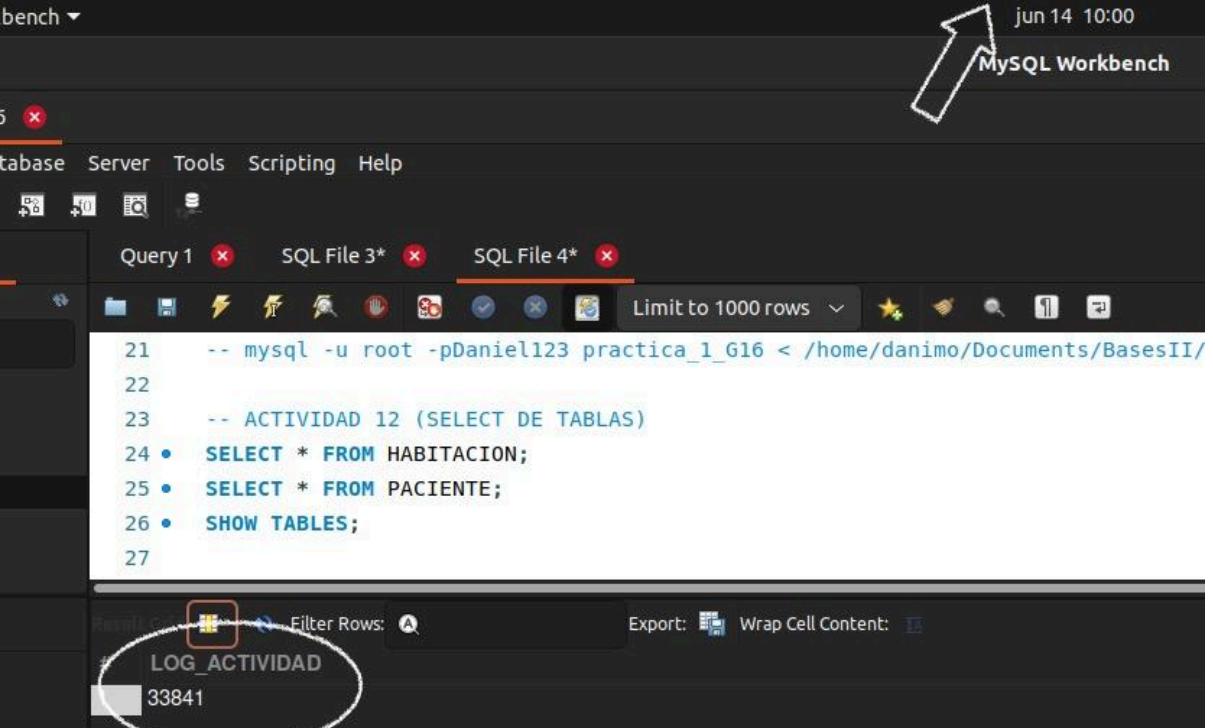
21 -- mysql -u root -pDaniel123 practica_1_G16 < /home/danimo/Documents/BasesII/Prac
22
23 -- ACTIVIDAD 12 (SELECT DE TABLAS)
24 • SELECT * FROM HABITACION;
25 • SELECT * FROM PACIENTE;
26 • SHOW TABLES;
27

```

The results pane shows the output of the last query, which is a table titled "PACIENTE". It has one row with the value 154184.

#	PACIENTE
1	154184

Imagen 13.7. SELECT COUNT(*) a la tabla PACIENTE.



The screenshot shows the MySQL Workbench interface. The title bar says "jun 14 10:00 MySQL Workbench". The menu bar includes Database, Server, Tools, Scripting, and Help. The toolbar has various icons for database management. Below the toolbar are three tabs: "Query 1", "SQL File 3*", and "SQL File 4*". The "Query 1" tab contains the same SQL code as in the previous screenshot:

```

21 -- mysql -u root -pDaniel123 practica_1_G16 < /home/danimo/Documents/BasesII/Prac
22
23 -- ACTIVIDAD 12 (SELECT DE TABLAS)
24 • SELECT * FROM HABITACION;
25 • SELECT * FROM PACIENTE;
26 • SHOW TABLES;
27

```

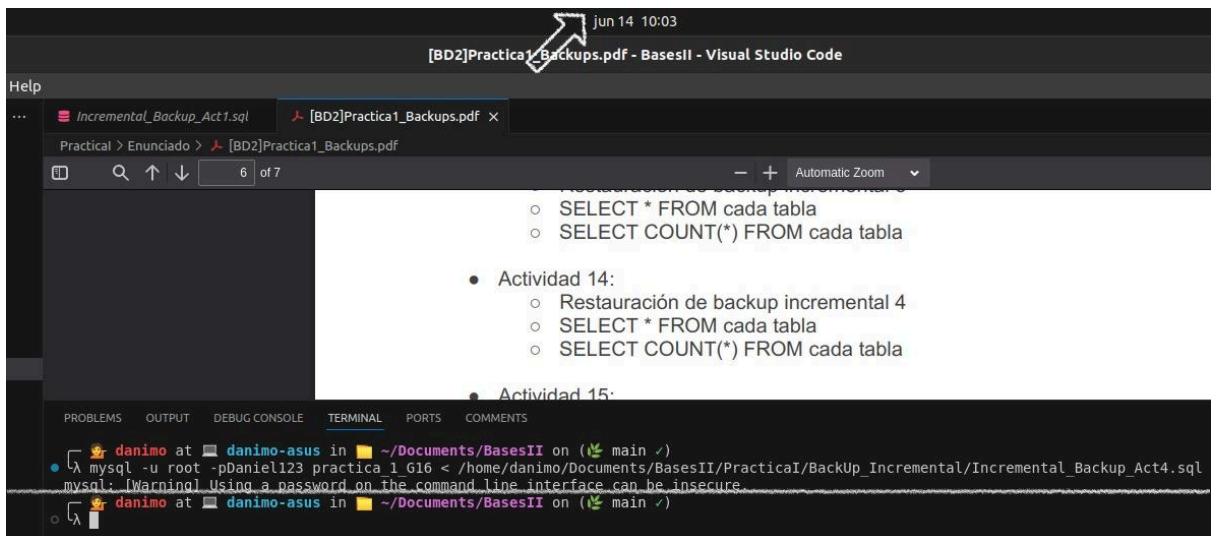
The results pane shows the output of the last query, which is a table titled "LOG_ACTIVIDAD". It has one row with the value 33841.

#	LOG_ACTIVIDAD
1	33841

Imagen 13.8. SELECT COUNT(*) a la tabla LOG_ACTIVIDAD.

Actividad 14

Para la segunda actividad en la restauración de backups incrementales. Tomando en cuenta que ya se debió de haber cargado el backup incremental anterior, ejecutamos el backup incremental 4 obtenido en la Actividad 4.



The screenshot shows a Visual Studio Code interface. At the top, there's a status bar with the date 'jun 14 10:03' and a tab labeled '[BD2]Practica1_Backups.pdf - BasesII - Visual Studio Code'. Below the status bar is a navigation bar with 'Help', 'Incremental_Backup_Act1.sql', '[BD2]Practica1_Backups.pdf X', 'Practical > Enunciado > [BD2]Practica1_Backups.pdf', and search/zoom controls. The main area contains two bulleted lists under 'Actividad 14:' and 'Actividad 15:', each with three options involving SELECT statements. At the bottom, a terminal window is open, showing a MySQL session. The command entered is 'mysql -u root -pDaniel123 practica_1_G16 < /home/danimo/Documents/BasesII/PracticaI/BackUp_Incremental/Incremental_Backup_Act4.sql'. A warning message from MySQL is visible: 'Warning! Using a password on the command line interface can be insecure.'

Imagen 14.1. Ejecución de comando para restaurar nuestro backup incremental 4.

Una vez realizado el backup podemos verificar que el backup se haya realizado ejecutando el comando SELECT en las tablas HABITACION, PACIENTE y LOG_ACTIVIDAD.

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a menu bar with 'base', 'Server', 'Tools', 'Scripting', and 'Help'. There are three tabs open: 'Query 1' (highlighted), 'SQL File 3*', and 'SQL File 4*'. The main area contains a SQL query:

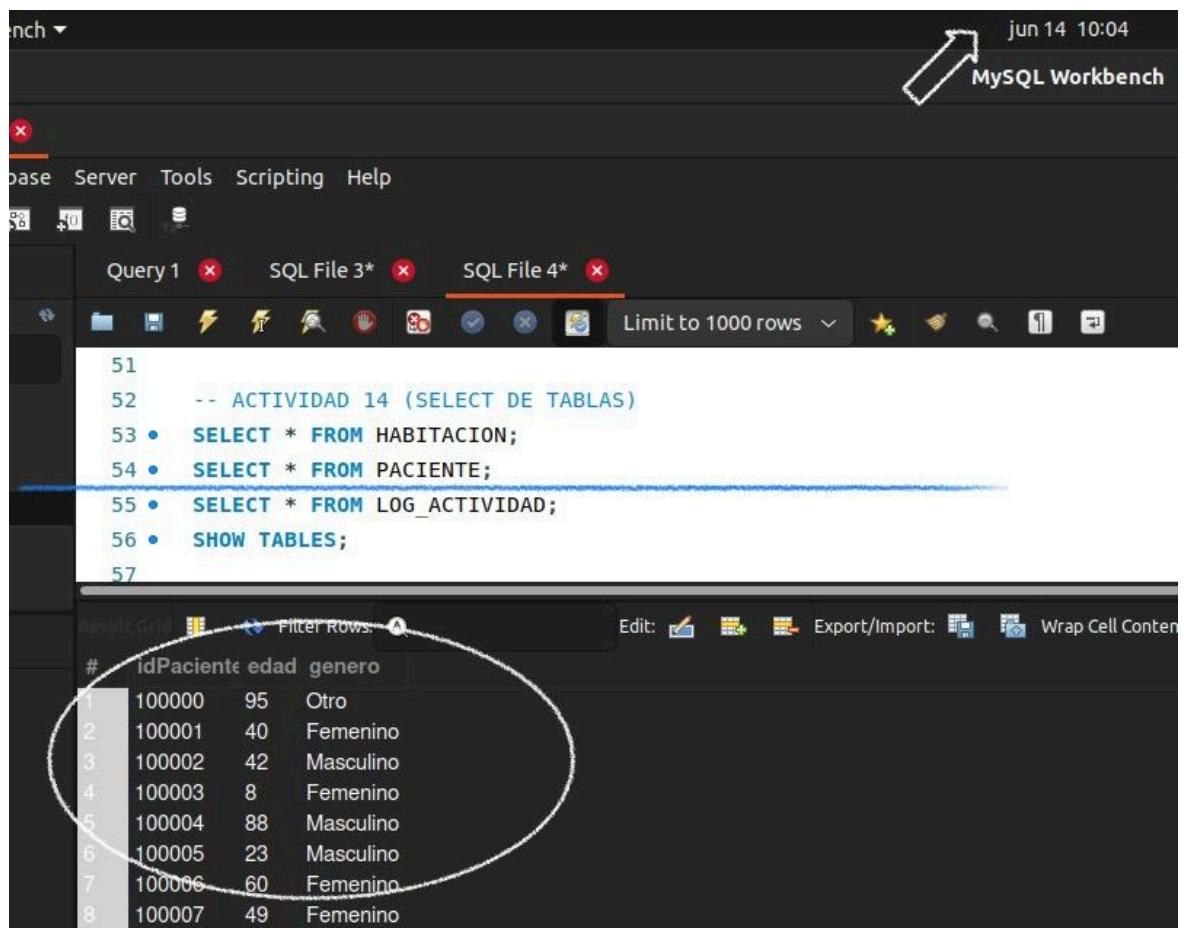
```
51
52 -- ACTIVIDAD 14 (SELECT DE TABLAS)
53 • SELECT * FROM HABITACION;
54 • SELECT * FROM PACIENTE;
55 • SELECT * FROM LOG_ACTIVIDAD;
56 • SHOW TABLES;
57
```

The 'Query 1' tab has a red arrow pointing to the 'SELECT * FROM HABITACION;' line. Below the query is a result grid titled 'Result Grid' with a circled 'Edit' button. The grid shows the following data:

#	iHabitacion	habitacion
1	1	Sala de exámenes 1
2	2	Sala de exámenes 2
3	3	Sala de exámenes 3
4	4	Sala de exámenes 4
5	5	Sala de imágenes 1
6	6	Sala de procedimientos 1
7	7	Sala de procedimientos 2
8	8	Sala de procedimientos 3
9	9	Sala de procedimientos 4

At the bottom right of the result grid, there are buttons for 'Edit', 'Filter Rows', 'Export/Import', and 'Wrap Cell Content'. The status bar at the bottom right shows 'jun 14 10:04'.

Imagen 14.2. SELECT a la tabla HABITACION.



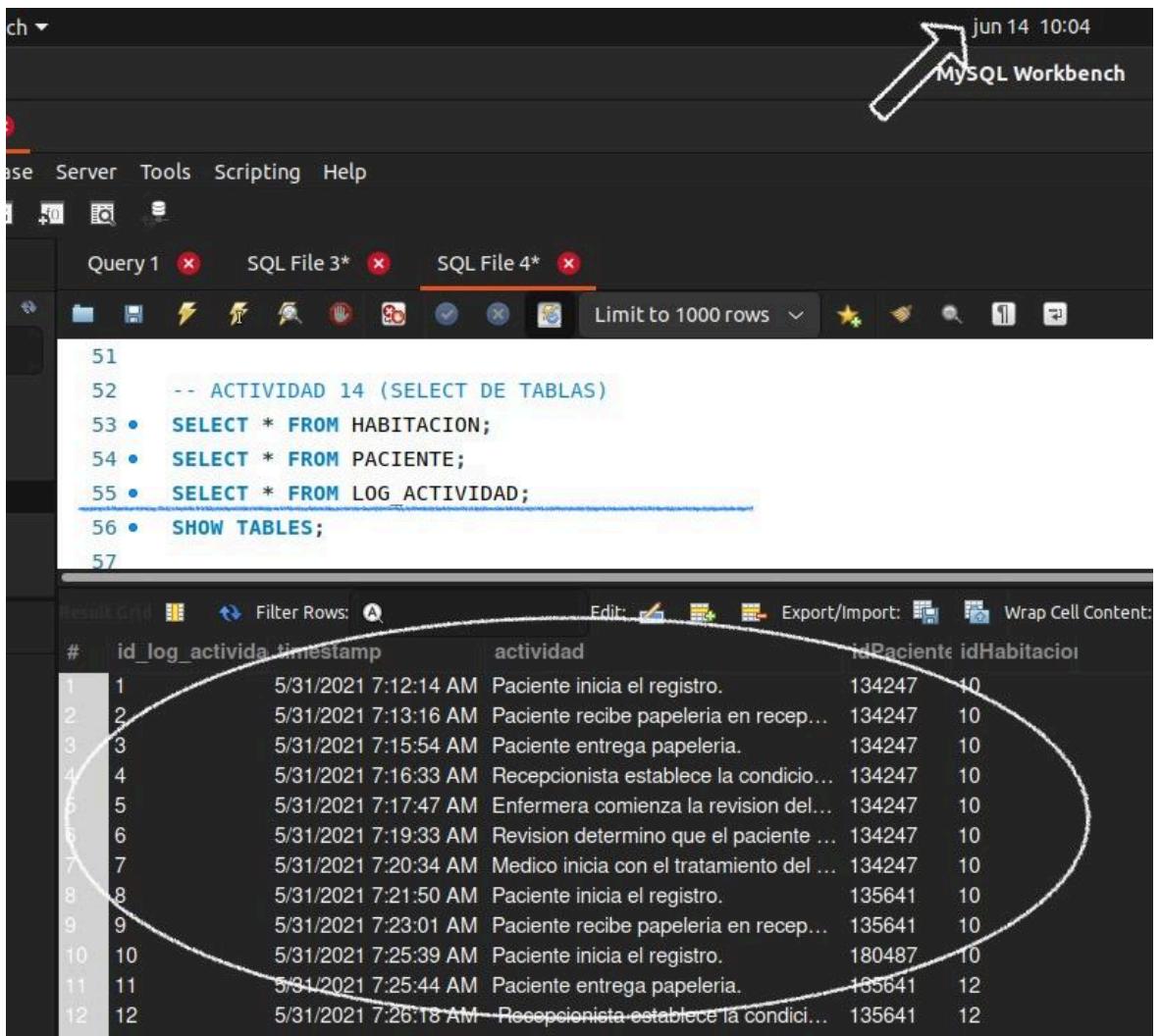
The screenshot shows the MySQL Workbench interface. At the top, there's a menu bar with 'File', 'Edit', 'Server', 'Tools', 'Scripting', and 'Help'. Below the menu is a toolbar with various icons. There are three tabs open: 'Query 1', 'SQL File 3*', and 'SQL File 4*'. The 'SQL File 4*' tab is active. The main area contains a SQL script:

```
51
52 -- ACTIVIDAD 14 (SELECT DE TABLAS)
53 • SELECT * FROM HABITACION;
54 • SELECT * FROM PACIENTE;
55 • SELECT * FROM LOG_ACTIVIDAD;
56 • SHOW TABLES;
57
```

Below the script is a results grid titled 'Result Grid' with a 'Filter Rows...' button. The grid has columns '#', 'idPaciente', 'edad', and 'genero'. The data is as follows:

#	idPaciente	edad	genero
1	100000	95	Otro
2	100001	40	Femenino
3	100002	42	Masculino
4	100003	8	Femenino
5	100004	88	Masculino
6	100005	23	Masculino
7	100006	60	Femenino
8	100007	49	Femenino

Imagen 14.3. SELECT a la tabla PACIENTE.



The screenshot shows the MySQL Workbench interface. In the top right corner, there is a timestamp "jun 14 10:04" and the application name "MySQL Workbench". The menu bar includes "Close", "Server", "Tools", "Scripting", and "Help". Below the menu is a toolbar with various icons. The main area has three tabs: "Query 1", "SQL File 3*", and "SQL File 4*". The "Query 1" tab is active and contains the following SQL code:

```

51
52 -- ACTIVIDAD 14 (SELECT DE TABLAS)
53 • SELECT * FROM HABITACION;
54 • SELECT * FROM PACIENTE;
55 • SELECT * FROM LOG_ACTIVIDAD;
56 • SHOW TABLES;
57

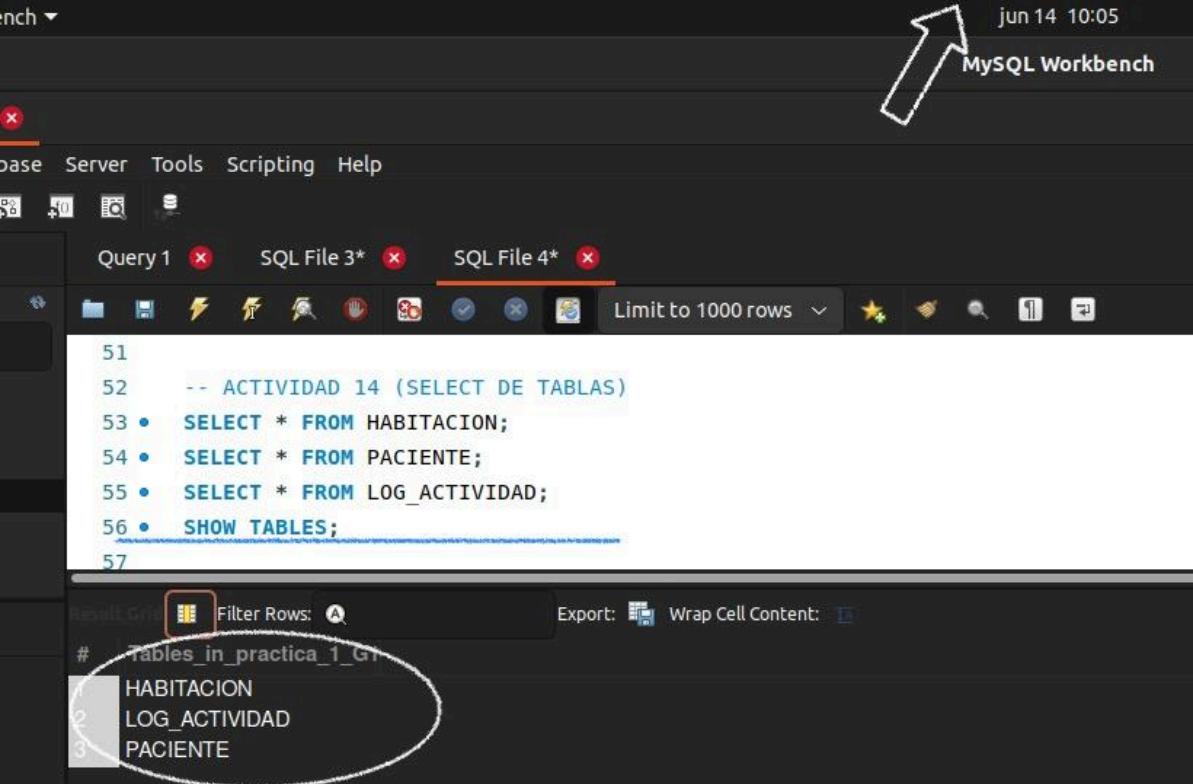
```

The "Result Grid" shows the output of the "SELECT * FROM LOG_ACTIVIDAD;" query. The columns are labeled "#", "id_log_actividad", "timestamp", "actividad", "idPaciente", and "idHabitacion". The data consists of 12 rows:

#	id_log_actividad	timestamp	actividad	idPaciente	idHabitacion
1	1	5/31/2021 7:12:14 AM	Paciente inicia el registro.	134247	10
2	2	5/31/2021 7:13:16 AM	Paciente recibe papeleria en recepcion...	134247	10
3	3	5/31/2021 7:15:54 AM	Paciente entrega papeleria.	134247	10
4	4	5/31/2021 7:16:33 AM	Repcionista establece la condicion...	134247	10
5	5	5/31/2021 7:17:47 AM	Enfermera comienza la revision del...	134247	10
6	6	5/31/2021 7:19:33 AM	Revision determino que el paciente ...	134247	10
7	7	5/31/2021 7:20:34 AM	Medico inicia con el tratamiento del ...	134247	10
8	8	5/31/2021 7:21:50 AM	Paciente inicia el registro.	135641	10
9	9	5/31/2021 7:23:01 AM	Paciente recibe papeleria en recepcion...	135641	10
10	10	5/31/2021 7:25:39 AM	Paciente inicia el registro.	180487	10
11	11	5/31/2021 7:25:44 AM	Paciente entrega papeleria.	135641	12
12	12	5/31/2021 7:26:18 AM	Repcionista establece la condicion...	135641	12

Imagen 14.4. SELECT a la tabla LOG_ACTIVIDAD.

Si ejecutamos el comando SHOW TABLES podremos notar que en esta ocasión no se ha creado ninguna tabla extra.



The screenshot shows the MySQL Workbench interface. In the top right corner, there is a watermark-like logo with the text "jun 14 10:05" and "MySQL Workbench". The main window has a dark theme. At the top, there is a menu bar with "Database", "Server", "Tools", "Scripting", and "Help". Below the menu is a toolbar with various icons. There are three tabs open in the center: "Query 1", "SQL File 3*", and "SQL File 4*". The "SQL File 4*" tab is active and contains the following SQL code:

```

51
52 -- ACTIVIDAD 14 (SELECT DE TABLAS)
53 • SELECT * FROM HABITACION;
54 • SELECT * FROM PACIENTE;
55 • SELECT * FROM LOG_ACTIVIDAD;
56 • SHOW TABLES;
57

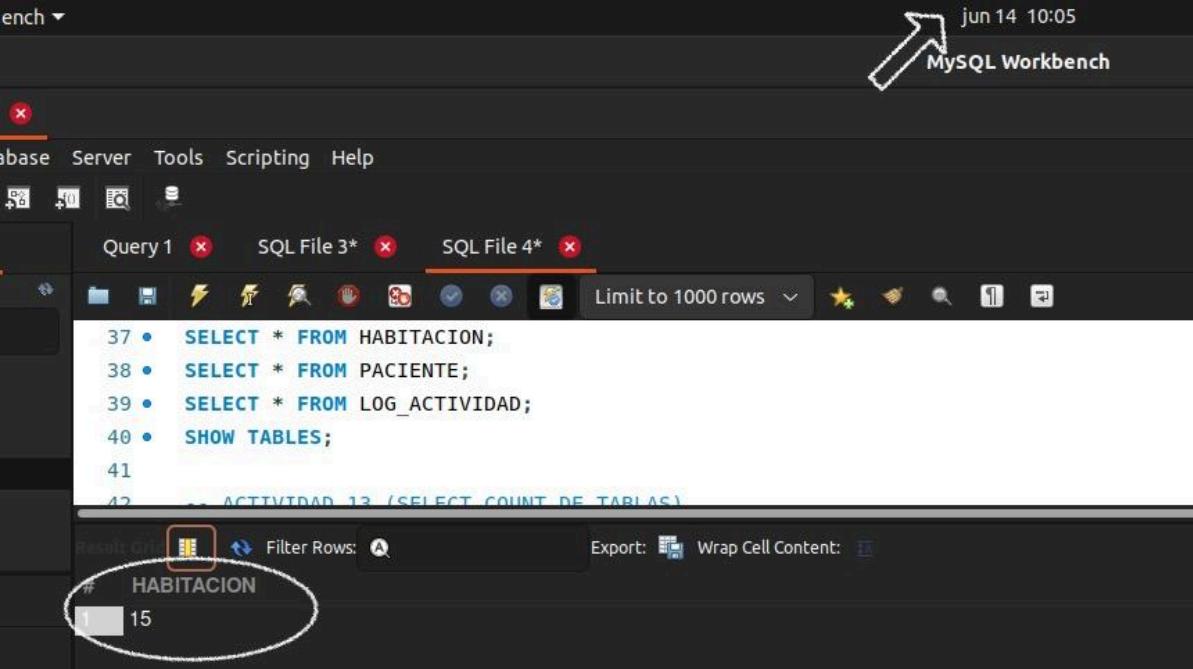
```

Below the code, the results pane shows the output of the "SHOW TABLES" command:

#	Tables_in_practica_1_Gr
1	HABITACION
2	LOG_ACTIVIDAD
3	PACIENTE

Imagen 14.5. SHOW TABLES a nuestra base de datos.

También podemos hacer uso del comando `SELECT COUNT(*)` para verificar que se hayan cargado los datos.



The screenshot shows the MySQL Workbench interface. In the top right corner, there is a watermark-like logo with the text "jun 14 10:05" and "MySQL Workbench". The main window has a dark theme. At the top, there is a menu bar with "Database", "Server", "Tools", "Scripting", and "Help". Below the menu is a toolbar with various icons. There are three tabs open in the center: "Query 1", "SQL File 3*", and "SQL File 4*". The "SQL File 4*" tab is active and contains the following SQL code:

```

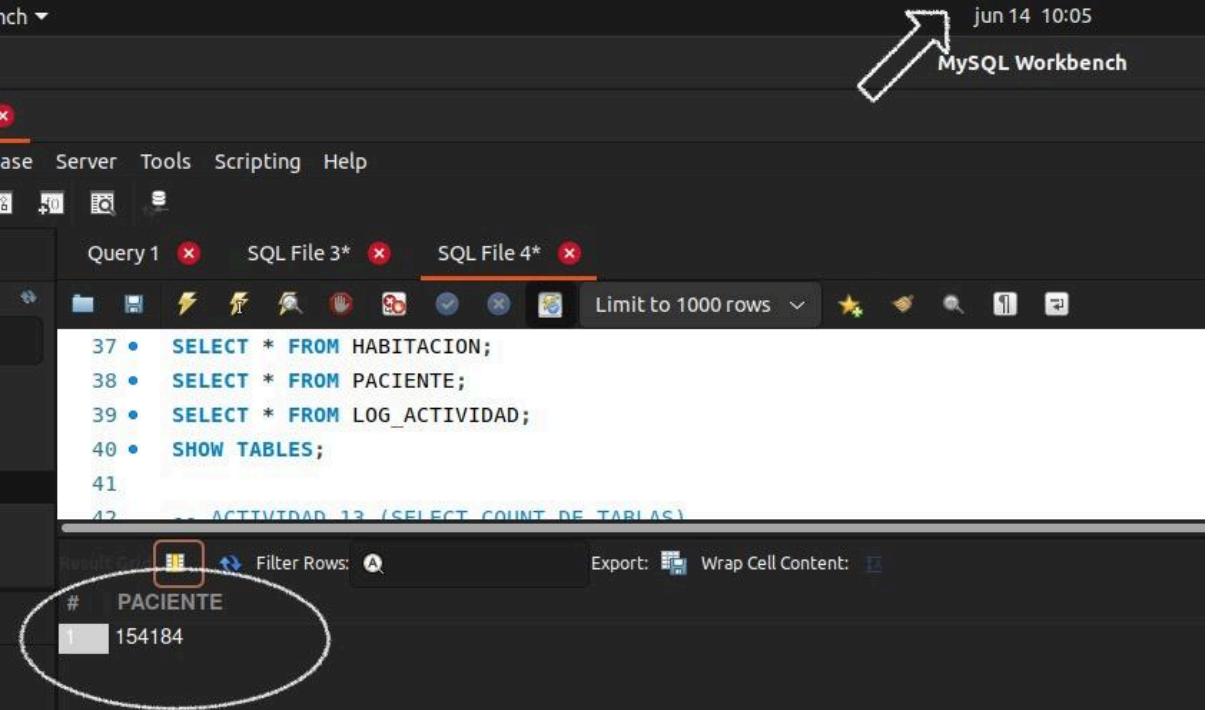
37 • SELECT * FROM HABITACION;
38 • SELECT * FROM PACIENTE;
39 • SELECT * FROM LOG_ACTIVIDAD;
40 • SHOW TABLES;
41
42 -- ACTIVIDAD 13 (SELECT COUNT DE TABLAS)

```

Below the code, the results pane shows the output of the `SELECT COUNT(*)` query on the "HABITACION" table:

#	
1	15

Imagen 14.6. `SELECT COUNT(*)` a la tabla HABITACION.



The screenshot shows the MySQL Workbench interface with the title bar "jun 14 10:05 MySQL Workbench". The menu bar includes "Database", "Server", "Tools", "Scripting", and "Help". Below the menu is a toolbar with various icons. The main area has three tabs: "Query 1", "SQL File 3*", and "SQL File 4*". The "Query 1" tab contains the following SQL code:

```

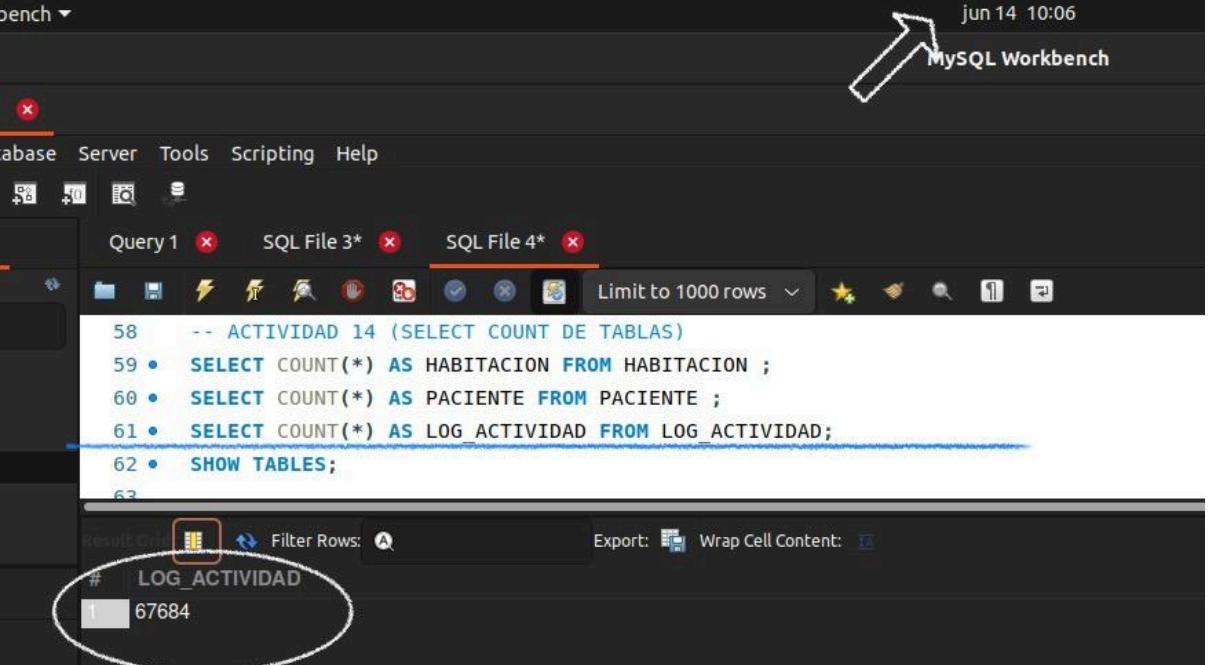
37 •  SELECT * FROM HABITACION;
38 •  SELECT * FROM PACIENTE;
39 •  SELECT * FROM LOG_ACTIVIDAD;
40 •  SHOW TABLES;
41
42 -- ACTIVIDAD 13 (SELECT COUNT DE TABLAS)

```

The results pane shows the output for the "PACIENTE" table:

#	PACIENTE
1	154184

Imagen 14.7. SELECT COUNT(*) a la tabla PACIENTE.



The screenshot shows the MySQL Workbench interface with the title bar "jun 14 10:06 MySQL Workbench". The menu bar includes "Database", "Server", "Tools", "Scripting", and "Help". Below the menu is a toolbar with various icons. The main area has three tabs: "Query 1", "SQL File 3*", and "SQL File 4*". The "Query 1" tab contains the following SQL code:

```

58 -- ACTIVIDAD 14 (SELECT COUNT DE TABLAS)
59 •  SELECT COUNT(*) AS HABITACION FROM HABITACION ;
60 •  SELECT COUNT(*) AS PACIENTE FROM PACIENTE ;
61 •  SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD;
62 •  SHOW TABLES;
63

```

The results pane shows the output for the "LOG_ACTIVIDAD" table:

#	LOG_ACTIVIDAD
1	67684

Imagen 14.8. SELECT COUNT(*) a la tabla LOG_ACTIVIDAD.

Actividad 15

Para la segunda actividad en la restauración de backups incrementales. Tomando en cuenta que ya se debió de haber cargado el backup incremental anterior, ejecutamos el backup incremental 5 obtenido en la Actividad 5.

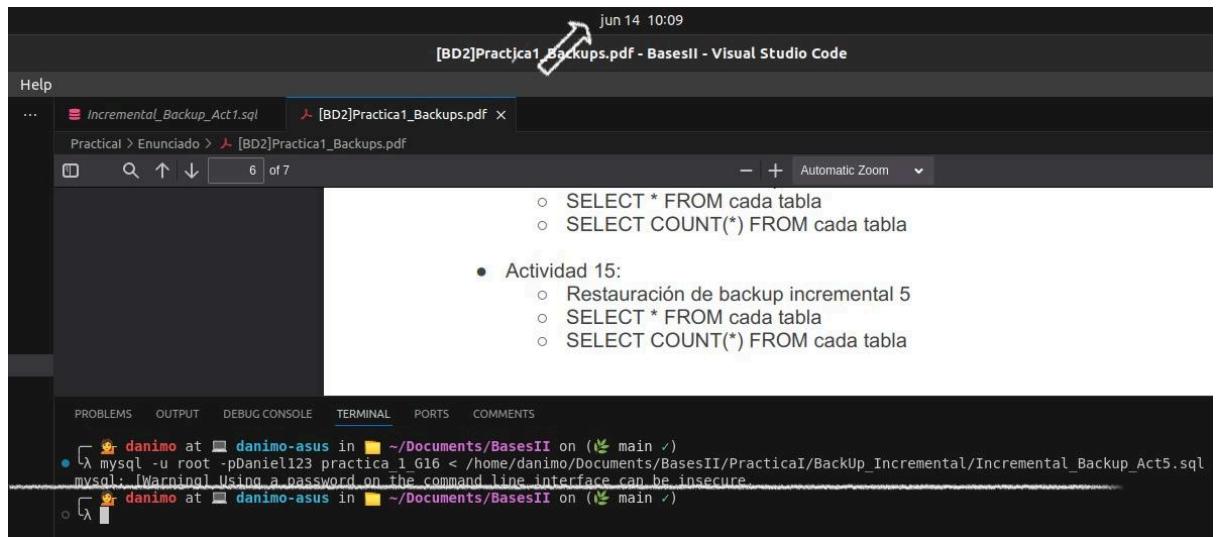


Imagen 15.1. Ejecución de comando para restaurar nuestro backup incremental 5.

Una vez realizado el backup podemos verificar que el backup se haya realizado ejecutando el comando SELECT en las tablas HABITACION, PACIENTE, LOG_ACTIVIDAD y LOG_HABITACION.

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a menu bar with Database, Server, Tools, Scripting, and Help. There are four tabs open in the main area: Query 1, SQL File 3*, and SQL File 4*. A red arrow points from the text "Imagen 15.2. SELECT a la tabla HABITACION." to the results grid. The results grid has columns labeled '#', 'idHabitacion' (with a red box around it), and 'habitación'. The data is as follows:

#	idHabitacion	habitación
1	1	Sala de exámenes 1
2	2	Sala de exámenes 2
3	3	Sala de exámenes 3
4	4	Sala de exámenes 4
5	5	Sala de imágenes 1
6	6	Sala de procedimientos 1
7	7	Sala de procedimientos 2
8	8	Sala de procedimientos 3
9	9	Sala de procedimientos 4
10	10	Recepción
11	11	Laboratorio
12	12	Estación de revisión 1
13	13	Estación de revisión 2
14	14	Estación de revisión 3
15	15	Estación de revisión 4

Imagen 15.2. SELECT a la tabla HABITACION.

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a menu bar with Database, Server, Tools, Scripting, and Help. There are three tabs open: Query 1, SQL File 3*, and SQL File 4*. The SQL File 4* tab contains the following code:

```
66
67 -- ACTIVIDAD 15 (SELECT DE TABLAS)
68 • SELECT * FROM HABITACION;
69 • SELECT * FROM PACIENTE;
70 • SELECT * FROM LOG_ACTIVIDAD;
71 • SELECT * FROM LOG_HABITACION;
72
```

Below the code is a result grid titled "Result Grid". It has columns labeled "#", "idPaciente", "edad", and "genero". The data is as follows:

#	idPaciente	edad	genero
1	100000	95	Otro
2	100001	40	Femenino
3	100002	42	Masculino
4	100003	8	Femenino
5	100004	88	Masculino
6	100005	23	Masculino
7	100006	60	Femenino
8	100007	49	Femenino
9	100008	44	Femenino
10	100009	55	Femenino
11	100010	28	Femenino

Imagen 15.3. SELECT a la tabla PACIENTE.

The screenshot shows the MySQL Workbench interface. At the top right, there is a timestamp 'jun 14 10:10' and a large hand-drawn style arrow pointing towards the results grid. The menu bar includes 'Database', 'Server', 'Tools', 'Scripting', and 'Help'. Below the menu is a toolbar with various icons. Three tabs are open: 'Query 1', 'SQL File 3*', and 'SQL File 4*'. The 'Query 1' tab contains the following SQL code:

```
66
67 -- ACTIVIDAD 15 (SELECT DE TABLAS)
68 • SELECT * FROM HABITACION;
69 • SELECT * FROM PACIENTE;
70 • SELECT * FROM LOG_ACTIVIDAD;
71 • SELECT * FROM LOG_HABITACION;
72
```

The results grid displays data from the 'LOG_ACTIVIDAD' table:

#	id_log_actividad	timestamp	actividad	idPaciente	idHabitacion
1	1	5/31/2021 7:12:14 AM	Paciente inicia el registro.	134247	10
2	2	5/31/2021 7:13:16 AM	Paciente recibe papeleria en recepcion.	134247	10
3	3	5/31/2021 7:15:54 AM	Paciente entrega papeleria.	134247	10
4	4	5/31/2021 7:16:33 AM	Repcionista establece la condicion.	134247	10
5	5	5/31/2021 7:17:47 AM	Enfermera comienza la revision del paciente.	134247	10
6	6	5/31/2021 7:19:33 AM	Revision determino que el paciente...	134247	10
7	7	5/31/2021 7:20:34 AM	Medico inicia con el tratamiento del paciente.	134247	10
8	8	5/31/2021 7:21:50 AM	Paciente inicia el registro.	135641	10
9	9	5/31/2021 7:23:01 AM	Paciente recibe papeleria en recepcion.	135641	10
10	10	5/31/2021 7:25:39 AM	Paciente inicia el registro.	180487	10
11	11	5/31/2021 7:25:44 AM	Paciente entrega papeleria.	135641	12
12	12	5/31/2021 7:26:18 AM	Repcionista establece la condicion.	135641	12

Imagen 15.4. SELECT a la tabla LOG_ACTIVIDAD.

The screenshot shows the MySQL Workbench interface. In the top right corner, there is a timestamp 'jun 14 10:10' and the text 'MySQL Workbench'. The main area displays a query editor with several tabs: 'Query 1', 'SQL File 3*', and 'SQL File 4*'. The 'Query 1' tab contains the following SQL code:

```
66
67 -- ACTIVIDAD 15 (SELECT DE TABLAS)
68 • SELECT * FROM HABITACION;
69 • SELECT * FROM PACIENTE;
70 • SELECT * FROM LOG_ACTIVIDAD;
71 • SELECT * FROM LOG_HABITACION;
```

The result grid below shows the data from the 'LOG_HABITACION' table. The columns are labeled '#', 'timestamp', 'status', and 'idHabitacion'. The data consists of 12 rows, each with a timestamp on '5/31/2021' and an ID between 6 and 14, indicating various room status changes.

#	timestamp	status	idHabitacion
1	5/31/2021 ...	Sala disponible.	6
2	5/31/2021 ...	Sala no disponible.	9
3	5/31/2021 ...	Inicia limpieza.	9
4	5/31/2021 ...	Sala disponible.	9
5	5/31/2021 ...	Sala no disponible.	13
6	5/31/2021 ...	Inicia limpieza.	13
7	5/31/2021 ...	Sala disponible.	13
8	5/31/2021 ...	Sala no disponible.	12
9	5/31/2021 ...	Inicia limpieza.	12
10	5/31/2021 ...	Sala disponible.	12
11	5/31/2021 ...	Sala no disponible.	14
12	5/31/2021 ...	Inicia limpieza.	14

Imagen 15.5. SELECT a la tabla LOG_HABITACION.

También podemos hacer uso del comando `SELECT COUNT(*)` para verificar que se hayan cargado los datos.

The screenshot shows the MySQL Workbench interface with the title bar "jun 14 10:11 MySQL Workbench". The menu bar includes Database, Server, Tools, Scripting, and Help. The toolbar has various icons for database management. There are four tabs: Query 1, SQL File 3*, and SQL File 4* (which is active). The SQL editor contains the following code:

```
72
73 -- ACTIVIDAD 15 (SELECT COUNT DE TABLAS)
74 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
75 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE ;
76 • SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD;
77 • SELECT COUNT(*) AS LOG_HABITACION FROM LOG_HABITACION;
78
```

The results grid shows the output of the last query:

#	HABITACION
1	15

Imagen 15.6. SELECT COUNT(*) a la tabla HABITACION.

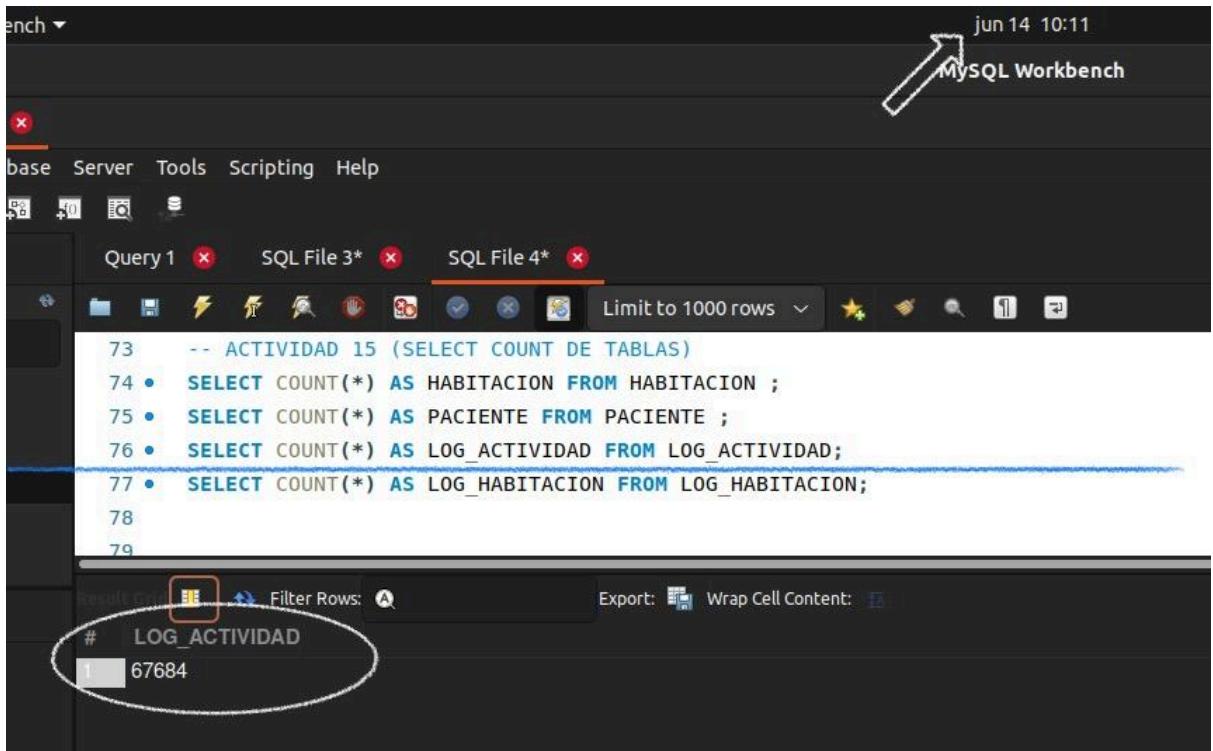
The screenshot shows the MySQL Workbench interface with the title bar "jun 14 10:11 MySQL Workbench". The menu bar includes Database, Server, Tools, Scripting, and Help. The toolbar has various icons for database management. There are four tabs: Query 1, SQL File 3*, and SQL File 4* (which is active). The SQL editor contains the same code as in Image 15.6:

```
73 -- ACTIVIDAD 15 (SELECT COUNT DE TABLAS)
74 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
75 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE ;
76 • SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD;
77 • SELECT COUNT(*) AS LOG_HABITACION FROM LOG_HABITACION;
78
79
```

The results grid shows the output of the second query:

#	PACIENTE
1	154184

Imagen 15.7. SELECT COUNT(*) a la tabla PACIENTE.



The screenshot shows the MySQL Workbench interface with a dark theme. In the top right corner, there is a timestamp 'jun 14 10:11' and the text 'MySQL Workbench'. The main area displays a SQL query window with several tabs: 'Query 1', 'SQL File 3*', and 'SQL File 4*'. The 'Query 1' tab contains the following SQL code:

```

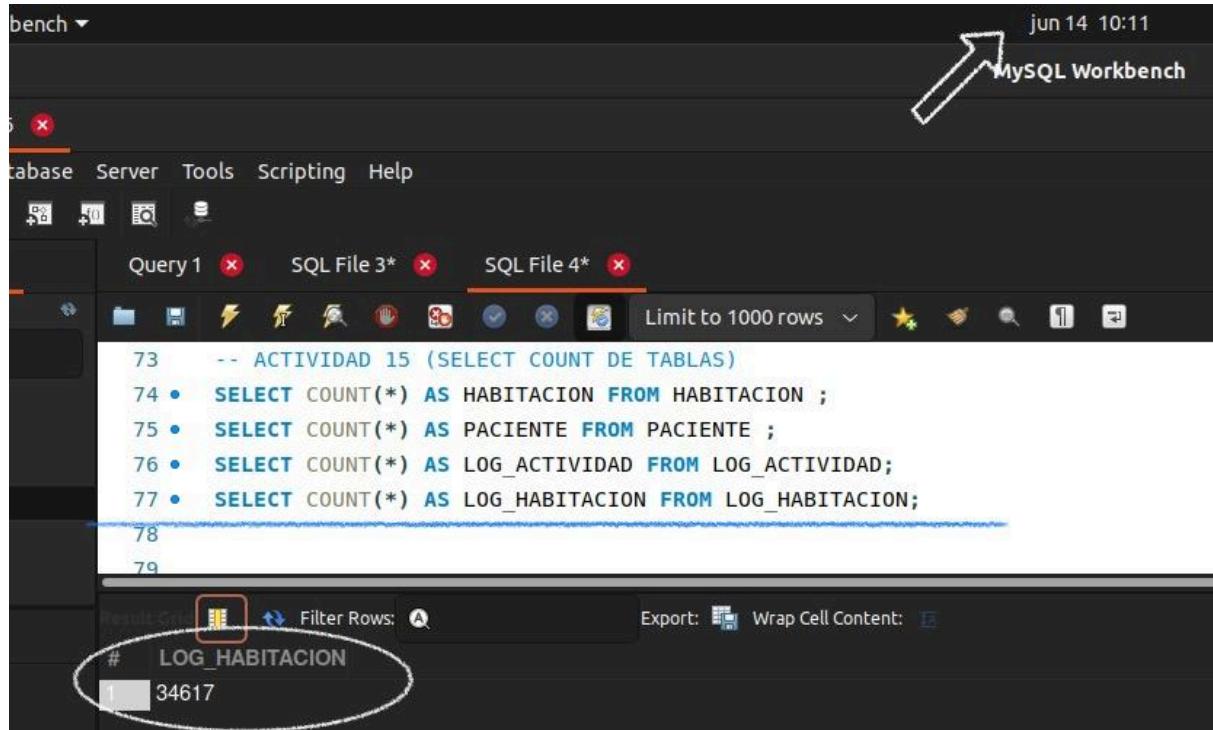
73 -- ACTIVIDAD 15 (SELECT COUNT DE TABLAS)
74 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
75 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE ;
76 • SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD;
77 • SELECT COUNT(*) AS LOG_HABITACION FROM LOG_HABITACION;
78
79

```

Below the code, the results are shown in a table with one row. The table has two columns: '# LOG_ACTIVIDAD'. The first row contains '# LOG_ACTIVIDAD' and the value '67684'. The entire row is highlighted with a blue selection bar.

#	LOG_ACTIVIDAD
1	67684

Imagen 15.8. SELECT COUNT(*) a la tabla LOG_ACTIVIDAD.



The screenshot shows the MySQL Workbench interface with a dark theme. In the top right corner, there is a timestamp 'jun 14 10:11' and the text 'MySQL Workbench'. The main area displays a SQL query window with three tabs: 'Query 1', 'SQL File 3*', and 'SQL File 4*'. The 'Query 1' tab contains the same SQL code as in Image 15.8:

```

73 -- ACTIVIDAD 15 (SELECT COUNT DE TABLAS)
74 • SELECT COUNT(*) AS HABITACION FROM HABITACION ;
75 • SELECT COUNT(*) AS PACIENTE FROM PACIENTE ;
76 • SELECT COUNT(*) AS LOG_ACTIVIDAD FROM LOG_ACTIVIDAD;
77 • SELECT COUNT(*) AS LOG_HABITACION FROM LOG_HABITACION;
78
79

```

Below the code, the results are shown in a table with one row. The table has two columns: '# LOG_HABITACION'. The first row contains '# LOG_HABITACION' and the value '34617'. The entire row is highlighted with a blue selection bar.

#	LOG_HABITACION
1	34617

Imagen 15.9. SELECT COUNT(*) a la tabla LOG_HABITACION.

Conclusiones

Erick Morales:

Durante este proyecto, comprendí la importancia de la planificación integral o comúnmente dicho “bien pensada” en la creación y restauración de backups, dado que son de gran ayuda ante diversas situaciones. La ejecución de backups completos e incrementales, seguida de una restauración exacta en la secuencia correcta, es esencial para garantizar la integridad y la recuperación completa de los datos en escenarios de pérdida. Esto me ha enseñado la importancia de mantener protocolos detallados y probados para manejar los datos de una manera más eficiente y tomando en cuenta la diferencia de cada tipo de backup, dado que cuando estábamos haciendo los incrementales pues no tenía acceso a toda la información, entonces el uso de estos es mas específico a un full backup, que nos da toda la información en un momento dado.

Leví Hernández:

Este proyecto me permitió apreciar el balance necesario entre eficiencia operativa y seguridad de los datos. Implementar backups incrementales de manera eficiente minimiza el uso de recursos y el tiempo de inactividad, mientras que asegurar una restauración confiable de backups completos protege la disponibilidad y seguridad de los datos críticos. Esta experiencia subraya cómo la gestión efectiva de recursos y la preparación para recuperaciones de emergencia son vitales para la continuidad operativa.

Luis Sánchez:

En el transcurso de la solución de ese proyecto, pude notar la importancia de realizar un enfoque equilibrado y adaptativo a la estrategia de implementación de backups al momento de construir una base de datos. En especial, cuando se está trabajando con data que es crítica y debe de estar disponible en todo momento, por lo que poder realizar una combinación estratégica entre backups complementarios e incrementales puede llegar a ofrecer un mejor balance entre la eficiencia operativa y la seguridad de datos que puede ofrecer una empresa. Implementar una política de backups que aproveche las ventajas de ambos tipos, ajustada a la frecuencia de cambios y el volumen de datos, permitirá asegurar la continuidad del negocio y la integridad de la información en cualquier escenario de recuperación.