

DEPARTAMENTO:	DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN	CARRERA:	INGENIERÍA EN SOFTWARE		
ASIGNATURA:	SISTEMAS DE BASE DE DATOS	PERIODO LECTIVO:	MAYO – SEPTIEMBRE 2023	NIVEL:	4to
DOCENTE:	ING. ELEANA JEREZ, MSc.	NRC:	9752	PRÁCTICA N°:	1
TEMA DE LA PRÁCTICA:	Base de datos empresarial MySQL/PostgreSQL				

INTRODUCCIÓN:

En el ámbito empresarial, el manejo eficiente de bases de datos es una tarea fundamental para garantizar la organización y seguridad de la información. En este trabajo, exploraremos el uso de dos plataformas de bases de datos ampliamente reconocidas en el mundo empresarial: MySQL y PostgreSQL.

Al iniciar nuestra experiencia con MySQL y PostgreSQL, nos enfrentamos a la necesidad de transferir archivos .dump, que contienen la estructura y los datos de una base de datos, desde el entorno local hacia las plataformas de bases de datos. Para lograr esto, recurrimos al uso del Command Prompt (CMD), una herramienta de la línea de comandos en sistemas operativos Windows, o la Terminal en sistemas operativos basados en Unix, como Linux y macOS.

Este proceso de transferencia de archivos .dump a través de la línea de comandos nos permitió crear bases de datos y restaurar su contenido de manera eficiente y precisa. A través de ejercicios prácticos descargados de las páginas oficiales de MySQL y PostgreSQL, adquirimos habilidades para manipular y administrar datos en entornos empresariales.

OBJETIVOS:

Objetivo General

Comparar el rendimiento y la eficiencia entre MySQL y PostgreSQL al cargar datos de pruebas y ejecutar consultas en un entorno de laboratorio.

Objetivos Específicos

- Cargar datos de prueba en una tabla de MySQL y otros datos similar en PostgreSQL.
- Realizar consultas de búsqueda básicas y pruebas en ambas bases de datos.
- Ejecutar consultas más complejas que involucren múltiples tablas y funciones de agregación.

EQUIPOS:

- Jose: PC (Hp Omen, I5 de 7 generación, 8 GB de ram, Gtx 1050).
- Martin: PC (Asus Tuf F17, I5 de 11ava, 16 GB de ram, Gtx 1650).
- Ricardo: PC (Dell, I7 de 11ava generación, 8 GB ram).
- Angelo: PC (Lenovo, I5 de 11ava generación, 8GB de ram, GTX 1050)

HERRAMIENTAS USADAS:

- Postgres 15 (pgAdmin4)
- MySQL
- PhpMyAdmin
- CMD
- DBeaver
- Xampp

INSTRUCCIONES:

EJERCICIO 1

Descargue e instale la base de datos “Employees” desde la documentación de MySQL <https://dev.mysql.com/doc/employee/en/employees-introduction.html>

1. Utilizando la estructura de la base de datos realice los inserts.
2. Si es el caso corregir el error del máximo de registros permitido aumentando la capacidad a 20 MB.
3. Ejecutar las pruebas de los archivos .sql y explicar a detalle su funcionalidad sobre la base de datos.
4. Realizar las siguientes consultas a la base de datos:
 - a. Obtener en una sola columna el nombre y el apellido de los empleados y en otra columna el salario.
 - b. Obtener en una sola columna el nombre y el apellido de los empleados y en otra columna el título.
 - c. Obtener en una sola columna el nombre y el apellido de los empleados y en otra columna el departamento.
 - d. Obtener en una sola columna el nombre y el apellido de los empleados y en otra columna el número de caracteres excepto espacios del nombre y el apellido del empleado juntos.
 - e. Obtener en una sola columna el nombre y el apellido de aquellos empleados que hayan sido despedidos el 21 de febrero de 1992 y en otra columna la fecha con el formato DD-MM-YYYY.
 - f. Obtener en una sola columna el nombre y el apellido de aquellos empleados que no pertenecían a un departamento en el 2001.
 - g. Obtener en una sola columna el nombre y apellido de aquellos empleados que ingresaron al departamento financiero antes de 1991.
 - h. Calcular el salario promedio de los empleados que son ingenieros “Engineer”
 - i. Obtener en una sola columna el nombre y apellido de aquellos empleados que tengan más de 27 años de experiencia en la empresa.
 - j. Obtener en una sola columna el nombre y apellido de aquellos empleados cuyos salarios sean menores a 3500.
 - k. Obtener en una sola columna el nombre y apellido y en otra columna el salario de aquellos empleados que sean ingenieros.
 - l. Obtener en una sola columna el nombre y apellido de aquellos empleados que ingresaron a la empresa antes del 1 de abril de 1991.
 - m. Obtener en una sola columna el nombre y apellido de aquellos empleados que salieron de la empresa después del 31 de diciembre de 1999 completando 8 años de servicio.

EJERCICIO 2

Instalar PostgreSQL

1. Utilizando la consola o pgAdmin o DBeaver trabaje con la base de datos “exercises” descargándosela de la página de recursos de PostgreSQL <https://pgexercises.com/>
2. Realice los ejercicios propuestos utilizando diferentes tablas y columnas a las usadas en los ejercicios.
 - a. Retrieve everything from a table.
 - b. Retrieve specific columns from a table.
 - c. Control which rows are retrieved.
 - d. Control which rows are retrieved – part 2.
 - e. Basic string searches.
3. Realice las siguientes consultas a la base de datos:
 - a. Mostrar una lista de las instalaciones con etiquetas de “barato” si el costo del mantenimiento es menor a 200 y “caro” si es mayor.
 - b. Mostrar en orden ascendente los 10 primeros apellidos de los miembros del club, el listado no debe contener duplicados.
 - c. Listar los starttime de las reservas de las canchas para la fecha 21-09-2012 y en otra columna el nombre de las instalaciones que corresponden a esas fechas, ordenar por tiempo ascendente.
 - d. Listar los miembros que han utilizado la cancha de tennis, incluir el nombre de la cancha y el nombre del miembro en una sola columna.
 - e. Listar los miembros quienes han recomedado a otro miembro. El resultado de la consulta no debe tener duplicados y deben estar ordenados primero por el apellido y luego por el primert nombre.
 - f. Borrar de la base de datos al mienbro cuyo id es 37 quien nunca ha realizado una reserva.

ACTIVIDADES POR DESARROLLAR EN EL LABORATORIO:

1. **Utilizando la estructura de la base de datos realice los inserts.**
 - a. Crear la base de datos:

```
Query OK, 0 rows affected (0.008 sec)

Query OK, 0 rows affected (0.007 sec)

Query OK, 0 rows affected (0.007 sec)

Query OK, 0 rows affected (0.006 sec)

Query OK, 0 rows affected (0.005 sec)

Query OK, 0 rows affected (0.000 sec)

+-----+
| INFO |
+-----+
| LOADING departments |
+-----+
1 row in set (0.000 sec)

ERROR at line 113 in file: 'C:\bd\employees.sql': Failed to open file 'load_departments.dump', error: 2

+-----+
| INFO |
+-----+
| LOADING employees |
+-----+
1 row in set (0.000 sec)

ERROR at line 115 in file: 'C:\bd\employees.sql': Failed to open file 'load_employees.dump', error: 2

+-----+
| INFO |
+-----+
| LOADING dept_emp |
+-----+
1 row in set (0.000 sec)

ERROR at line 117 in file: 'C:\bd\employees.sql': Failed to open file 'load_dept_emp.dump', error: 2

+-----+
| INFO |
+-----+
| LOADING dept_manager |
+-----+
1 row in set (0.000 sec)

ERROR at line 119 in file: 'C:\bd\employees.sql': Failed to open file 'load_dept_manager.dump', error: 2

+-----+
| INFO |
+-----+
| LOADING titles |
+-----+
```

b. Realizar un testeo para comprar si está bien creada la base de datos:

```
Query OK, 1 row affected (0.001 sec)

Query OK, 0 rows affected (0.006 sec)

+-----+
| table_name | found_records | found_crc |
+-----+
| departments | 0 | 
| dept_emp | 0 | 
| dept_manager | 0 | 
| employees | 0 | 
| salaries | 0 | 
| titles | 0 | 
+-----+
6 rows in set (0.000 sec)

+-----+
| table_name | records_match | crc_match |
+-----+
| departments | not ok | not ok |
| dept_emp | not ok | not ok |
| dept_manager | not ok | not ok |
| employees | not ok | not ok |
| salaries | not ok | not ok |
| titles | not ok | not ok |
+-----+
6 rows in set (0.001 sec)

Query OK, 0 rows affected (0.001 sec)

Query OK, 0 rows affected (0.000 sec)

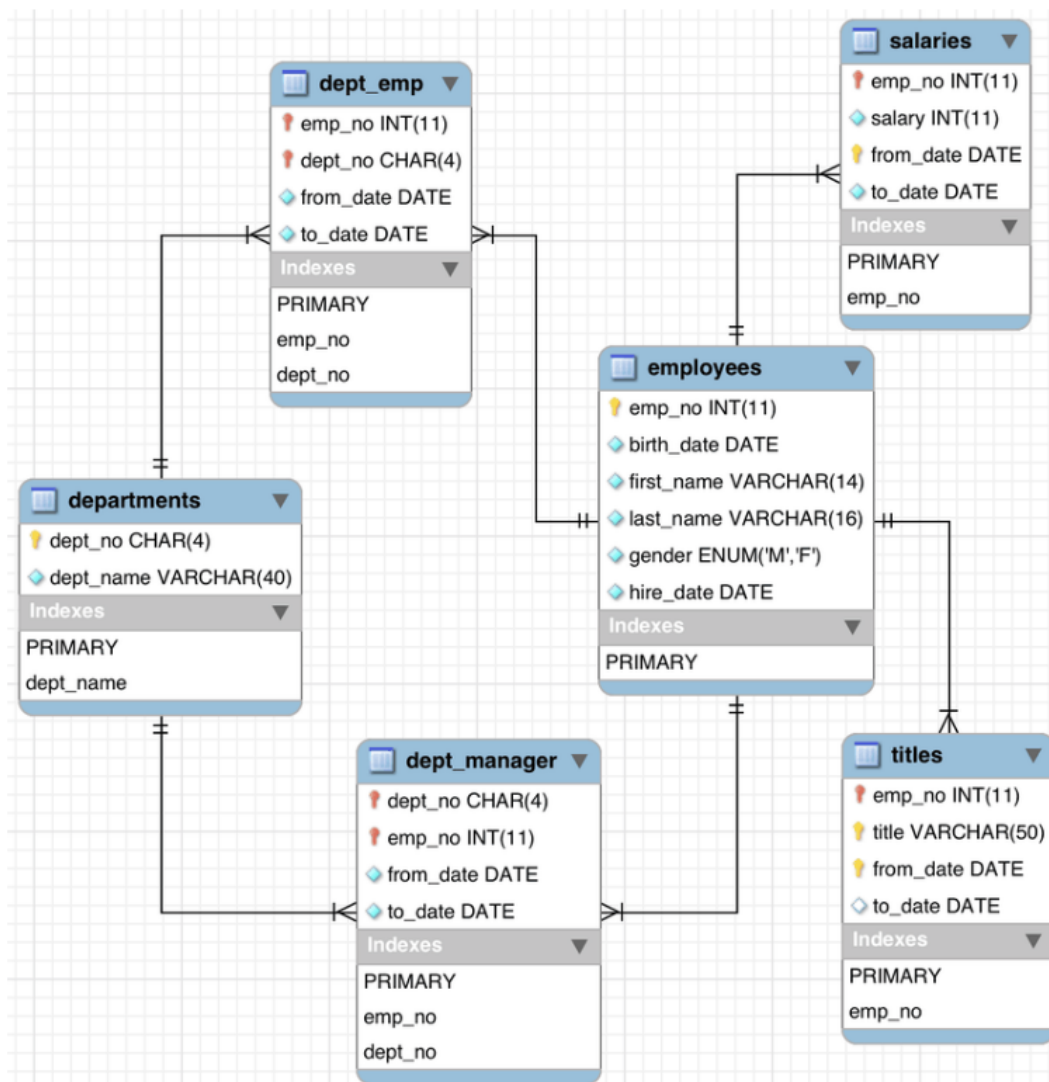
+-----+
| computation_time |
+-----+
| 00:00:00 |
+-----+
1 row in set (0.001 sec)

Query OK, 0 rows affected (0.011 sec)

+-----+
| summary | result |
+-----+
| CRC | FAIL |
| count | FAIL |
+-----+
2 rows in set (0.000 sec)
```



c. Para realizar los inserts. nos guiaremos en el diagrama E-R físico





d. Realizar los inserts en la tabla employees

```
release changes
MariaDB [employees]> SOURCE C:\bd\load_employees.dump
Query OK, 17944 rows affected (0.144 sec)
Records: 17944 Duplicates: 0 Warnings: 0

Query OK, 17938 rows affected (0.138 sec)
Records: 17938 Duplicates: 0 Warnings: 0

Query OK, 17953 rows affected (0.125 sec)
Records: 17953 Duplicates: 0 Warnings: 0

Query OK, 17947 rows affected (0.130 sec)
Records: 17947 Duplicates: 0 Warnings: 0

Query OK, 17948 rows affected (0.117 sec)
Records: 17948 Duplicates: 0 Warnings: 0

Query OK, 17648 rows affected (0.409 sec)
Records: 17648 Duplicates: 0 Warnings: 0

Query OK, 17642 rows affected (0.125 sec)
Records: 17642 Duplicates: 0 Warnings: 0

Query OK, 17638 rows affected (0.113 sec)
Records: 17638 Duplicates: 0 Warnings: 0

Query OK, 17653 rows affected (0.167 sec)
Records: 17653 Duplicates: 0 Warnings: 0

Query OK, 17650 rows affected (0.123 sec)
Records: 17650 Duplicates: 0 Warnings: 0

Query OK, 17636 rows affected (0.368 sec)
Records: 17636 Duplicates: 0 Warnings: 0

Query OK, 17642 rows affected (0.107 sec)
Records: 17642 Duplicates: 0 Warnings: 0

Query OK, 17646 rows affected (0.118 sec)
Records: 17646 Duplicates: 0 Warnings: 0

Query OK, 17642 rows affected (0.114 sec)
Records: 17642 Duplicates: 0 Warnings: 0

Query OK, 17651 rows affected (0.147 sec)
Records: 17651 Duplicates: 0 Warnings: 0
```



e. Realizar los inserts en la tabla titles.

```
MariaDB [employees]> SOURCE C:\bd\load_titles.dump
Query OK, 21707 rows affected (0.171 sec)
Records: 21707 Duplicates: 0 Warnings: 0

Query OK, 21700 rows affected (0.130 sec)
Records: 21700 Duplicates: 0 Warnings: 0

Query OK, 21687 rows affected (0.189 sec)
Records: 21687 Duplicates: 0 Warnings: 0

Query OK, 21694 rows affected (0.120 sec)
Records: 21694 Duplicates: 0 Warnings: 0

Query OK, 21695 rows affected (0.137 sec)
Records: 21695 Duplicates: 0 Warnings: 0

Query OK, 21700 rows affected (0.130 sec)
Records: 21700 Duplicates: 0 Warnings: 0

Query OK, 21326 rows affected (0.146 sec)
Records: 21326 Duplicates: 0 Warnings: 0

Query OK, 21257 rows affected (0.149 sec)
Records: 21257 Duplicates: 0 Warnings: 0

Query OK, 21245 rows affected (0.132 sec)
Records: 21245 Duplicates: 0 Warnings: 0

Query OK, 21266 rows affected (0.134 sec)
Records: 21266 Duplicates: 0 Warnings: 0

Query OK, 21279 rows affected (0.121 sec)
Records: 21279 Duplicates: 0 Warnings: 0

Query OK, 21251 rows affected (0.120 sec)
Records: 21251 Duplicates: 0 Warnings: 0

Query OK, 21270 rows affected (0.135 sec)
Records: 21270 Duplicates: 0 Warnings: 0

Query OK, 21261 rows affected (0.139 sec)
Records: 21261 Duplicates: 0 Warnings: 0
```



f. Realizar los inserts en la tabla salaries parte 1

```
MariaDB [employees]> SOURCE C:\bd\load_salaries1.dump
Query OK, 25526 rows affected (0.152 sec)
Records: 25526 Duplicates: 0 Warnings: 0

Query OK, 25527 rows affected (0.124 sec)
Records: 25527 Duplicates: 0 Warnings: 0

Query OK, 25529 rows affected (0.166 sec)
Records: 25529 Duplicates: 0 Warnings: 0

Query OK, 25527 rows affected (0.132 sec)
Records: 25527 Duplicates: 0 Warnings: 0

Query OK, 25528 rows affected (0.134 sec)
Records: 25528 Duplicates: 0 Warnings: 0

Query OK, 25528 rows affected (0.194 sec)
Records: 25528 Duplicates: 0 Warnings: 0

Query OK, 25527 rows affected (0.143 sec)
Records: 25527 Duplicates: 0 Warnings: 0

Query OK, 25525 rows affected (0.121 sec)
Records: 25525 Duplicates: 0 Warnings: 0

Query OK, 25531 rows affected (0.127 sec)
Records: 25531 Duplicates: 0 Warnings: 0

Query OK, 25528 rows affected (0.130 sec)
Records: 25528 Duplicates: 0 Warnings: 0

Query OK, 25528 rows affected (0.131 sec)
Records: 25528 Duplicates: 0 Warnings: 0

Query OK, 25528 rows affected (0.140 sec)
Records: 25528 Duplicates: 0 Warnings: 0

Query OK, 25527 rows affected (0.197 sec)
Records: 25527 Duplicates: 0 Warnings: 0

Query OK, 25529 rows affected (0.151 sec)
Records: 25529 Duplicates: 0 Warnings: 0

Query OK, 25525 rows affected (0.151 sec)
Records: 25525 Duplicates: 0 Warnings: 0

Query OK, 25530 rows affected (0.132 sec)
Records: 25530 Duplicates: 0 Warnings: 0
```




g. Realizar los inserts en la tabla salaries parte 2

```
MariaDB [employees]> SOURCE C:\bd\load_salaries2.dump
Query OK, 24922 rows affected (0.122 sec)
Records: 24922 Duplicates: 0 Warnings: 0

Query OK, 24923 rows affected (0.138 sec)
Records: 24923 Duplicates: 0 Warnings: 0

Query OK, 24922 rows affected (0.147 sec)
Records: 24922 Duplicates: 0 Warnings: 0

Query OK, 24920 rows affected (0.126 sec)
Records: 24920 Duplicates: 0 Warnings: 0

Query OK, 24921 rows affected (0.134 sec)
Records: 24921 Duplicates: 0 Warnings: 0

Query OK, 24922 rows affected (0.219 sec)
Records: 24922 Duplicates: 0 Warnings: 0

Query OK, 24923 rows affected (0.131 sec)
Records: 24923 Duplicates: 0 Warnings: 0

Query OK, 24922 rows affected (0.186 sec)
Records: 24922 Duplicates: 0 Warnings: 0

Query OK, 24922 rows affected (0.153 sec)
Records: 24922 Duplicates: 0 Warnings: 0

Query OK, 24922 rows affected (0.124 sec)
Records: 24922 Duplicates: 0 Warnings: 0

Query OK, 24922 rows affected (0.193 sec)
Records: 24922 Duplicates: 0 Warnings: 0

Query OK, 24921 rows affected (0.121 sec)
Records: 24921 Duplicates: 0 Warnings: 0

Query OK, 24920 rows affected (0.146 sec)
Records: 24920 Duplicates: 0 Warnings: 0

Query OK, 24920 rows affected (0.120 sec)
Records: 24920 Duplicates: 0 Warnings: 0

Query OK, 24918 rows affected (0.134 sec)
Records: 24918 Duplicates: 0 Warnings: 0

Query OK, 24920 rows affected (0.124 sec)
Records: 24920 Duplicates: 0 Warnings: 0
```



h. Realizar los inserts en la tabla salaries parte 3

```
MariaDB [employees]> SOURCE C:\bd\load_salaries3.dump
Query OK, 24917 rows affected (0.119 sec)
Records: 24917 Duplicates: 0 Warnings: 0

Query OK, 24920 rows affected (0.135 sec)
Records: 24920 Duplicates: 0 Warnings: 0

Query OK, 24919 rows affected (0.124 sec)
Records: 24919 Duplicates: 0 Warnings: 0

Query OK, 24923 rows affected (0.134 sec)
Records: 24923 Duplicates: 0 Warnings: 0

Query OK, 24921 rows affected (0.181 sec)
Records: 24921 Duplicates: 0 Warnings: 0

Query OK, 24920 rows affected (0.126 sec)
Records: 24920 Duplicates: 0 Warnings: 0

Query OK, 24920 rows affected (0.173 sec)
Records: 24920 Duplicates: 0 Warnings: 0

Query OK, 24920 rows affected (0.119 sec)
Records: 24920 Duplicates: 0 Warnings: 0

Query OK, 24920 rows affected (0.150 sec)
Records: 24920 Duplicates: 0 Warnings: 0

Query OK, 24919 rows affected (0.133 sec)
Records: 24919 Duplicates: 0 Warnings: 0

Query OK, 24921 rows affected (0.130 sec)
Records: 24921 Duplicates: 0 Warnings: 0

Query OK, 24919 rows affected (0.197 sec)
Records: 24919 Duplicates: 0 Warnings: 0

Query OK, 24921 rows affected (0.151 sec)
Records: 24921 Duplicates: 0 Warnings: 0

Query OK, 24922 rows affected (0.124 sec)
Records: 24922 Duplicates: 0 Warnings: 0

Query OK, 24922 rows affected (0.155 sec)
Records: 24922 Duplicates: 0 Warnings: 0

Query OK, 24919 rows affected (0.122 sec)
Records: 24919 Duplicates: 0 Warnings: 0
```

i. Realizar los inserts en la tabla departaments

```
MariaDB [employees]> SOURCE C:\bd\load_departments.dump
Query OK, 9 rows affected (0.006 sec)
Records: 9 Duplicates: 0 Warnings: 0
```



j. Realizar los inserts en la tabla dept_manager

```
MariaDB [employees]> SOURCE C:\bd\load_dept_manager.dump
Query OK, 24 rows affected (0.005 sec)
Records: 24 Duplicates: 0 Warnings: 0
```

k. Realizar los inserts en la tabla dept_emp

```
MariaDB [employees]> SOURCE C:\bd\load_dept_emp.dump
Query OK, 24940 rows affected (0.235 sec)
Records: 24940 Duplicates: 0 Warnings: 0

Query OK, 24940 rows affected (0.173 sec)
Records: 24940 Duplicates: 0 Warnings: 0

Query OK, 24940 rows affected (0.240 sec)
Records: 24940 Duplicates: 0 Warnings: 0

Query OK, 24934 rows affected (0.270 sec)
Records: 24934 Duplicates: 0 Warnings: 0

Query OK, 24360 rows affected (0.178 sec)
Records: 24360 Duplicates: 0 Warnings: 0

Query OK, 24360 rows affected (0.278 sec)
Records: 24360 Duplicates: 0 Warnings: 0

Query OK, 24360 rows affected (0.239 sec)
Records: 24360 Duplicates: 0 Warnings: 0

Query OK, 24360 rows affected (0.190 sec)
Records: 24360 Duplicates: 0 Warnings: 0

Query OK, 24360 rows affected (0.192 sec)
Records: 24360 Duplicates: 0 Warnings: 0

Query OK, 24360 rows affected (0.252 sec)
Records: 24360 Duplicates: 0 Warnings: 0

Query OK, 24360 rows affected (0.204 sec)
Records: 24360 Duplicates: 0 Warnings: 0

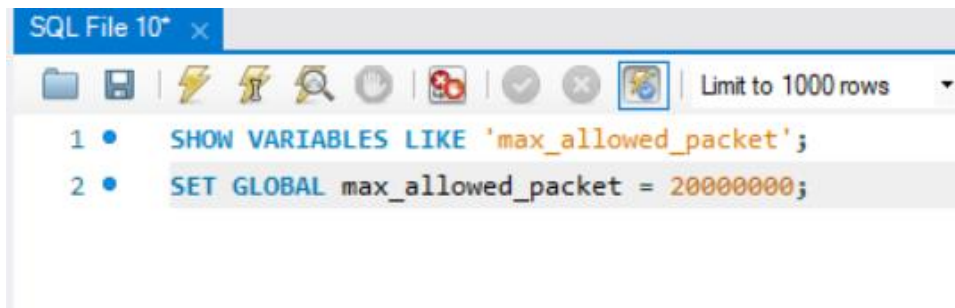
Query OK, 24360 rows affected (0.190 sec)
Records: 24360 Duplicates: 0 Warnings: 0

Query OK, 24360 rows affected (0.194 sec)
Records: 24360 Duplicates: 0 Warnings: 0

Query OK, 12609 rows affected (0.112 sec)
Records: 12609 Duplicates: 0 Warnings: 0
```



2. Si es el caso corregir el error del máximo de registros permitido aumentando la capacidad a 20 MB.



3. Ejecutar las pruebas de los archivos .sql y explicar a detalle su funcionalidad sobre la base de datos.

- a. Ejecución de la primera prueba

```
mysql> source /media/test_employees_md5.sql
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
+-----+
| INFO                                     |
+-----+
| TESTING INSTALLATION                   |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected, 2 warnings (0.10 sec)

Query OK, 0 rows affected (0.18 sec)

Query OK, 0 rows affected (0.06 sec)

Query OK, 6 rows affected (0.01 sec)
Records: 6 Duplicates: 0 Warnings: 0

+-----+
| table_name | expected_records | expected_crc |
+-----+
| departments | 9                | d1af5e170d2d1591d776d5638d71fc5f |
| dept_emp    | 331603           | ccf6fe516f990bdaa49713fc478701b7 |
| dept_manager | 24              | 8720e2f0853ac9096b689c14664f847e |
| employees   | 300024          | 4ec56ab5ba37218d187cf6ab09ce1aa1 |
| salaries    | 2844047         | fd220654e95aea1b169624ffe3fca934 |
| titles      | 443308          | bfa016c472df68e70a03facafa1bc0a8 |
+-----+
6 rows in set (0.00 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.05 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 300024 rows affected, 1 warning (10.00 sec)
Records: 300024 Duplicates: 0 Warnings: 1

Query OK, 1 row affected (0.65 sec)
```



6 rows in set (0.00 sec)

table_name	records_match	crc_match
departments	OK	ok
dept_emp	OK	ok
dept_manager	OK	ok
employees	OK	ok
salaries	OK	ok
titles	OK	ok

6 rows in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.00 sec)

computation_time
00:01:15

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.04 sec)

summary	result
CRC	OK
count	OK

2 rows in set (0.00 sec)



```
Records: 9 Duplicates: 0 Warnings: 1

Query OK, 1 row affected (0.05 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 24 rows affected, 1 warning (0.05 sec)
Records: 24 Duplicates: 0 Warnings: 1

Query OK, 1 row affected (0.01 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 331603 rows affected, 1 warning (8.69 sec)
Records: 331603 Duplicates: 0 Warnings: 1

Query OK, 1 row affected (0.69 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 443308 rows affected, 1 warning (8.89 sec)
Records: 443308 Duplicates: 0 Warnings: 1

Query OK, 1 row affected (0.35 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 2844047 rows affected, 1 warning (44.45 sec)
Records: 2844047 Duplicates: 0 Warnings: 1

Query OK, 1 row affected (1.41 sec)

Query OK, 0 rows affected (0.09 sec)
```

table_name	found_records	found_crc
departments	9	d1af5e170d2d1591d776d5638d71fc5f
dept_emp	331603	ccf6fe516f990bdaa49713fc478701b7
dept_manager	24	8720e2f0853ac9096b689c14664f847e
employees	300024	4ec56ab5ba37218d187cf6ab09ce1aa1
salaries	2844047	fd220654e95aea1b169624ffe3fca934
titles	443308	bfa016c472df68e70a03facafa1bc0a8

b. Ejecución de la segunda prueba

```
Records: 9 Duplicates: 0 Warnings: 1

Query OK, 1 row affected (0.05 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 24 rows affected, 1 warning (0.05 sec)
Records: 24 Duplicates: 0 Warnings: 1

Query OK, 1 row affected (0.01 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 331603 rows affected, 1 warning (8.69 sec)
Records: 331603 Duplicates: 0 Warnings: 1

Query OK, 1 row affected (0.69 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 443308 rows affected, 1 warning (8.89 sec)
Records: 443308 Duplicates: 0 Warnings: 1

Query OK, 1 row affected (0.35 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 2844047 rows affected, 1 warning (44.45 sec)
Records: 2844047 Duplicates: 0 Warnings: 1

Query OK, 1 row affected (1.41 sec)

Query OK, 0 rows affected (0.09 sec)
```

table_name	found_records	found_crc
departments	9	d1af5e170d2d1591d776d5638d71fc5f
dept_emp	331603	ccf6fe516f990bdaa49713fc478701b7
dept_manager	24	8720e2f0853ac9096b689c14664f847e
employees	300024	4ec56ab5ba37218d187cf6ab09ce1aa1
salaries	2844047	fd220654e95aea1b169624ffe3fca934
titles	443308	bfa016c472df68e70a03facafa1bc0a8



```
6 rows in set (0.00 sec)
```

table_name	records_match	crc_match
departments	OK	ok
dept_emp	OK	ok
dept_manager	OK	ok
employees	OK	ok
salaries	OK	ok
titles	OK	ok

```
6 rows in set (0.00 sec)
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
Query OK, 0 rows affected (0.00 sec)
```

computation_time
00:01:15

```
1 row in set (0.00 sec)
```

```
Query OK, 0 rows affected (0.04 sec)
```

summary	result
CRC	OK
count	OK

```
2 rows in set (0.00 sec)
```

c. Utilidad de las pruebas

Lo que realiza ambas pruebas es ver que la cantidad de datos ingresados en la base sean igual al esperado, además que esos datos se encuentren en buen estado. Las pruebas garantizan la integridad y precisión de los datos, por lo que el ejercicio descargado de la fuente oficial fortalece nuestra comprensión y habilidades en estas bases de datos empresariales, asegurando su correcto funcionamiento en entornos profesionales.

4. Consultas

- Obtener en una sola columna el nombre y apellido de los empleados y en otra columna los salarios.



departments salaries titles SQL File 5* x

Limit to 1000 rows

```
1 • SELECT CONCAT(e.first_name, ' ', e.last_name) AS nombre_completo, s.salary
2 FROM employees e JOIN salaries s ON e.emp_no = s.emp_no;
3
4 /*SELECT CONCAT(e.first_name, ' ', e.last_name) AS nombre_completo, t.title
5 FROM employees e JOIN titles t ON e.emp_no = t.emp_no;*/
```

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

	nombre_completo	salary
▶	Georgi Facello	60117
	Georgi Facello	62102
	Georgi Facello	66074
	Georgi Facello	66596
	Georgi Facello	66961
	Georgi Facello	71046
	Georgi Facello	74333
	Georgi Facello	75286
	Georgi Facello	75994
	Georgi Facello	76884
	Georgi Facello	80013
	Georgi Facello	81025
	Georgi Facello	81097
	Georgi Facello	84917
	Georgi Facello	85112
	Georgi Facello	85097
	Georgi Facello	88958
	Bezael Simmel	65828
	Bezael Simmel	65909

Result 13 x

- b. Obtener en una sola columna el nombre y apellido de los empleados y en otra columna el título.

departments salaries titles SQL File 5* x

Limit to 1000 rows

```
3
4 • SELECT CONCAT(e.first_name, ' ', e.last_name) AS nombre_completo, t.title
5 FROM employees e JOIN titles t ON e.emp_no = t.emp_no;
6
7 /*SELECT CONCAT(e.first_name, ' ', e.last_name) AS nombre_completo, d.dept_name
```

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

	nombre_completo	title
▶	Georgi Facello	Senior Engineer
	Bezael Simmel	Staff
	Parto Bamford	Senior Engineer
	Chirstian Koblick	Engineer
	Chirstian Koblick	Senior Engineer
	Kyoichi Maliniak	Senior Staff
	Kyoichi Maliniak	Staff
	Anneke Preusig	Senior Engineer
	Tzvetan Zielinski	Senior Staff
	Tzvetan Zielinski	Staff
	Saniya Kalloufi	Assistant Engi...
	Sumant Peac	Assistant Engi...
	Sumant Peac	Engineer
	Sumant Peac	Senior Engineer
	Duangkaew Pive...	Engineer
	Mary Sluis	Staff
	Patricio Bridgland	Engineer
	Patricio Bridgland	Senior Engineer
	Eberhardt Terkki	Senior Staff

Result 10 x



- c. Obtener en una sola columna el nombre y apellido de los empleados y en otra columna el nombre del departamento.

SQL File 5*

```
6
7 • SELECT CONCAT(e.first_name, ' ', e.last_name) AS nombre_completo, d.dept_name
8 FROM employees e JOIN dept_emp de ON e.emp_no = de.emp_no
9 JOIN departments d ON d.dept_no = d.dept_no;
10
```

Result Grid

nombre_completo	dept_name
Georgi Facello	Customer Service
Georgi Facello	Development
Georgi Facello	Finance
Georgi Facello	Human Resources
Georgi Facello	Marketing
Georgi Facello	Production
Georgi Facello	Quality Management
Georgi Facello	Research
Georgi Facello	Sales
Bezalet Simmel	Customer Service
Bezalet Simmel	Development
Bezalet Simmel	Finance
Bezalet Simmel	Human Resources
Bezalet Simmel	Marketing
Bezalet Simmel	Production
Bezalet Simmel	Quality Management
Bezalet Simmel	Research
Bezalet Simmel	Sales
Parto Bamford	Customer Service

Result 11

- d. Obtener en una sola columna el nombre y apellido de los empleados y en otra columna el número de los caracteres sin contar los espacios.

SQL File 5*

```
10
11 • SELECT CONCAT(first_name, ' ', last_name) AS nombre_completo,
12 LENGTH(REPLACE(CONCAT(first_name, last_name), ' ', '')) AS caracteres_sin_espacios
13 FROM employees;
14
```

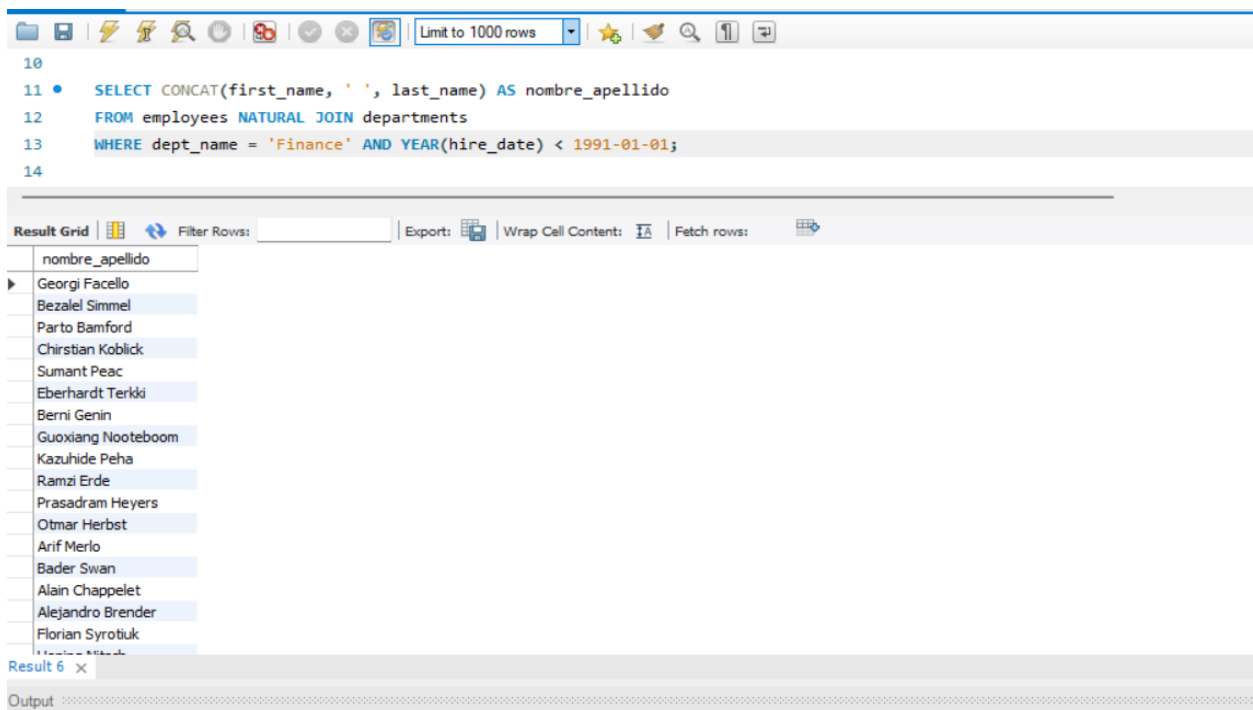
Result Grid

nombre_completo	caracteres_sin_espacios
Georgi Facello	13
Bezalet Simmel	13
Parto Bamford	12
Chirstian Koblick	16
Kyoichi Maliniak	15
Anneke Preusig	13
Tzvetan Zielinski	16
Saniya Kalloufi	14
Sumant Peac	10
Duangkaew Piveteau	17
Mary Sluis	9
Patricio Bridgland	17
Eberhardt Terkki	15
Berni Genin	10
Guoxiang Nootboom	17
Kazuhiro Cappelletti	19
Cristinel Bouloucos	18
Kazuhide Peha	12
Lillian Haddadi	14

Result 12

[illegible]

- g. Obtener en una sola columna el nombre y apellido de aquellos empleados que ingresaron al departamento financiero antes de 1991.



```

10
11 • SELECT CONCAT(first_name, ' ', last_name) AS nombre_apellido
12 FROM employees NATURAL JOIN departments
13 WHERE dept_name = 'Finance' AND YEAR(hire_date) < 1991-01-01;
14

```

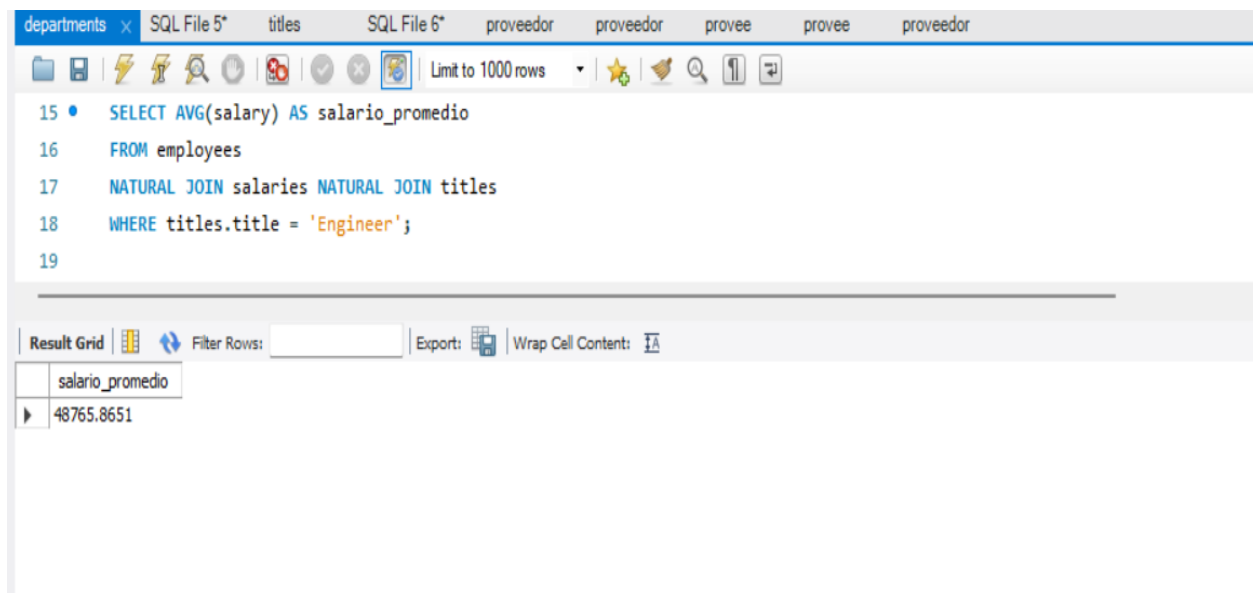
Result Grid

nombre_apellido
Georgi Facello
Bezael Simmel
Parto Bamford
Chirstian Koblick
Sumant Peac
Eberhardt Terkki
Berni Genin
Guoxiang Nootboom
Kazuhide Peha
Ramzi Erde
Prasadram Heyers
Otmar Herbst
Arif Merlo
Bader Swan
Alain Chapelet
Alejandro Brender
Florian Syrotiuk

Result 6 x

Output

- h. Calcular el salario promedio de los empleados que son ingenieros “Engineer”.



```

15 • SELECT AVG(salary) AS salario_promedio
16 FROM employees
17 NATURAL JOIN salaries NATURAL JOIN titles
18 WHERE titles.title = 'Engineer';
19

```

Result Grid

salario_promedio
48765.8651



- i. Obtener en una sola columna el nombre y apellido de aquellos empleados que tengan más de 27 años de experiencia en la empresa.

```
20
21 • SELECT CONCAT(e.first_name, ' ', e.last_name) AS nombre_completo_empleado
22 FROM employees e JOIN dept_emp d ON e.emp_no = d.emp_no
23 WHERE (d.to_date - d.from_date)>=27;
24
```

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

nombre_completo_empleado
Georgi Facello
Bezael Simmel
Parto Bamford
Christian Koblick
Kyoichi Maliniak
Anneke Preusig
Tzvetan Zielinski
Saniya Kalloufi
Sumant Peac
Duangkaew Piveteau
Duangkaew Piveteau
Mary Sluis
Patricio Bridgland
Eberhardt Terkki
Berni Genin
Guoxiang Nootboom
Kazuhiro Cappelletti

Result 8 x

- j. Obtener en una sola columna el nombre y apellido de aquellos empleados cuyos salarios sean menores a 3500.

Salario como tal de \$3500

```
23
24 /*Consulta j
25 (Obtener en una sola columna el nombre y apellido de aquellos empleados cuyos salarios sean menores a 3500)*/
26 • SELECT CONCAT(e.first_name, ' ', e.last_name) AS "Nombre Empleado"
27 FROM employees e JOIN salaries s ON e.emp_no = s.emp_no
28 WHERE s.salary<3500;
29
```

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

Nombre Empleado



Salario dividido para 12 meses

The screenshot shows a SQL query in the 'SQL Editor' window. The query is as follows:

```
23  
24 /*Consulta j  
25 (Obtener en una sola columna el nombre y apellido de aquellos empleados cuyos salarios sean menores a  
26 • SELECT CONCAT(e.first_name, ' ', e.last_name) AS "Nombre Empleado"  
27 FROM employees e JOIN salaries s ON e.emp_no = s.emp_no  
28 WHERE (s.salary/12)<3500;  
29
```

The 'Result Grid' shows the following data:

Nombre Empleado
Parto Bamford
Chirstian Koblick
Anneke Preusig
Patricio Bridgland
Patricio Bridgland
Eberhardt Terkki
Eberhardt Terkki
Eberhardt Terkki
Eberhardt Terkki
Guoxiang Nooteboom
Mayuko Warwick
Mayuko Warwick
Shahaf Famili

- k. Obtener en una sola columna el nombre y apellido y en otra columna el salario de aquellos empleados que sean ingenieros.

The screenshot shows a SQL query in the 'SQL Editor' window. The query is as follows:

```
30  
31 • SELECT CONCAT(e.first_name, ' ', e.last_name) AS nombre_completo, s.salary AS salary, t.title  
32 FROM employees e  
33 JOIN titles t ON e.emp_no = t.emp_no  
34 JOIN salaries s ON e.emp_no = s.emp_no
```

The 'Result Grid' shows the following data:

nombre_completo	salary	title
Chirstian Koblick	40054	Engineer
Chirstian Koblick	42283	Engineer
Chirstian Koblick	42542	Engineer
Chirstian Koblick	46065	Engineer
Chirstian Koblick	48271	Engineer
Chirstian Koblick	50594	Engineer
Chirstian Koblick	52119	Engineer
Chirstian Koblick	54693	Engineer
Chirstian Koblick	58326	Engineer
Chirstian Koblick	60770	Engineer
Chirstian Koblick	62566	Engineer
Chirstian Koblick	64340	Engineer
Chirstian Koblick	67096	Engineer
Chirstian Koblick	69722	Engineer
Chirstian Koblick	70698	Engineer
Chirstian Koblick	74057	Engineer
Sumant Peac	60929	Engineer
Sumant Peac	60929	Engineer



1. Obtener en una sola columna el nombre y apellido de aquellos empleados que ingresaron a la empresa antes del 1 de abril de 1991.

SQL Developer interface showing a query and its results. The query is:

```
37 SELECT CONCAT(e.first_name, ' ', e.last_name) AS nombre_completo, e.hire_date
38 FROM employees e
39 WHERE e.hire_date < '1991-04-01';
40
41 /*SELECT concat(first_name, ' ', last_name) AS Empleado
```

The results are displayed in a table with two columns: nombre_completo and hire_date. The first 18 rows are shown, with the last row being the 18th result.

nombre_completo	hire_date
Georgi Facello	1986-06-26
Bezael Simmel	1985-11-21
Paro Bamford	1986-08-28
Chirstian Koblick	1986-12-01
Kyoichi Maliniak	1989-09-12
Anneke Preusig	1989-06-02
Tzvetan Zielinski	1989-02-10
Sumant Peac	1985-02-18
Duangkaew Piveteau	1989-08-24
Mary Sluis	1990-01-22
Eberhardt Terkki	1985-10-20
Berni Genin	1987-03-11
Guoxiang Nooteboom	1987-07-02
Kazuhide Peha	1987-04-03
Mayuko Warwick	1991-01-26
Ramzi Erde	1988-02-10
Bojan Montemayor	1989-12-17

- m. Obtener en una sola columna el nombre y apellido de aquellos empleados que salieron de la empresa después del 31 de diciembre de 1999 completando 8 años de servicio.

SQL Developer interface showing a query and its results. The query is:

```
41 SELECT concat(first_name, ' ', last_name) AS Empleado
42 FROM employees, dept_emp WHERE to_date < '1999-12-31'
43 AND datediff(to_date, hire_date) >= 8*365
44 AND employees.emp_no = dept_emp.emp_noj
```

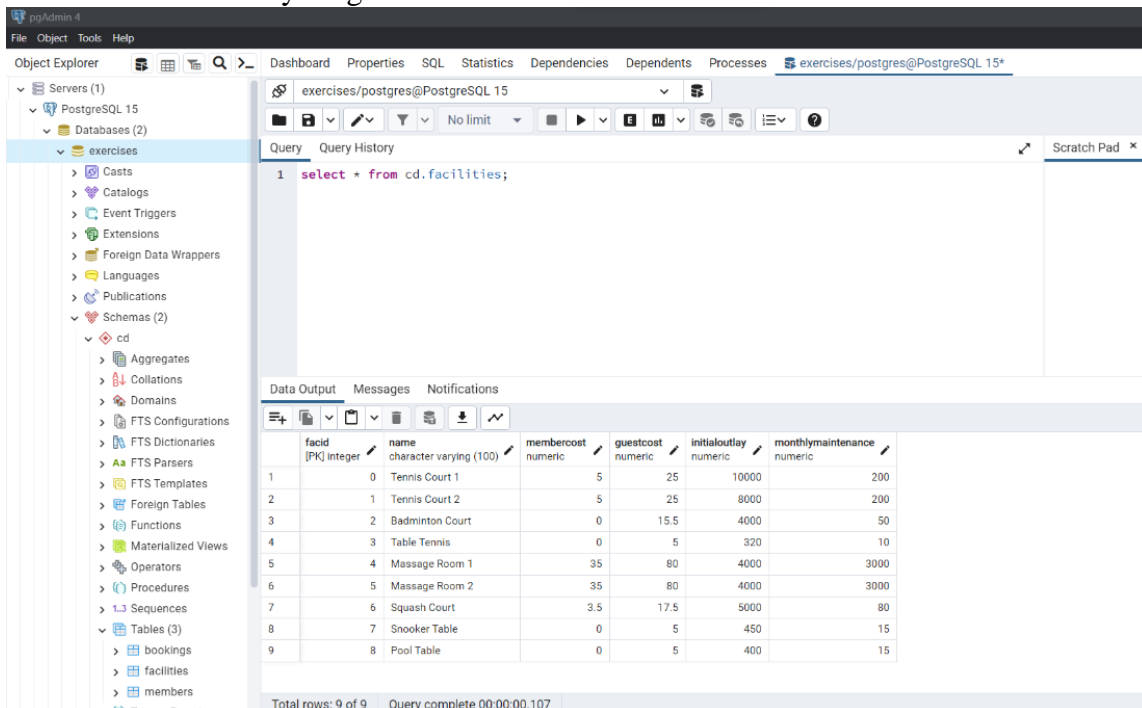
The results are displayed in a table with one column: Empleado. The first 19 rows are shown, with the last row being the 19th result.

Empleado
Prasadram Heyers
Otmar Herbst
Bader Swan
Reuven Garigliano
Premal Baek
Kendra Hofting
Arumugam Ossenbruggen
Lunjin Giveon
Dayanand Czap
Dayanand Czap
Domenick Peltason
Zissis Pintelas
Xiadong Perry
Moriyoshi Merey
Shigehito Kropatsch
Mohua Falck
Nevio Ritcey

EJERCICIO 2

Instalar PostgreSQL

1. Utilizando la consola o pgAdmin o DBeaver trabaje con la base de datos “exercises” descargándosela de la página de recursos de PostgreSQL <https://pgexercises.com/>
2. Realice los ejercicios propuestos utilizando diferentes tablas y columnas a las usadas en los ejercicios.
 - a. Retrieve everything from a table.

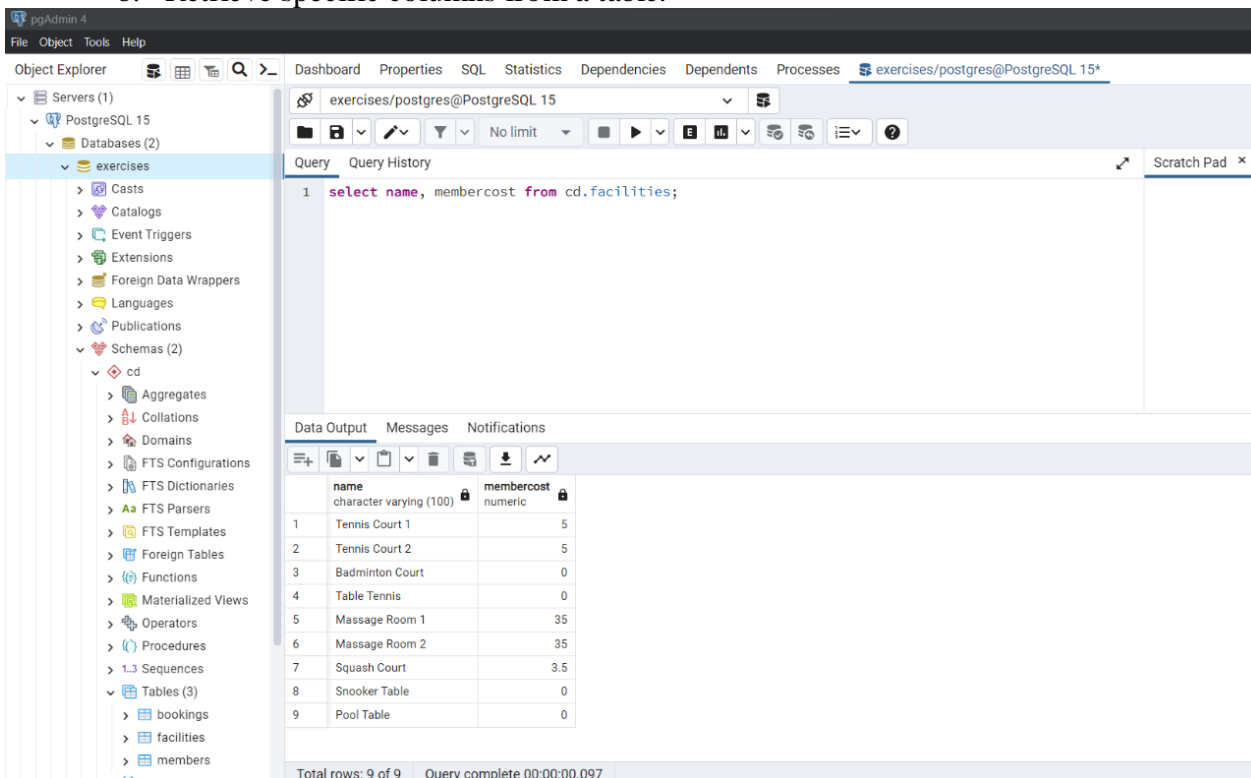


The screenshot shows the pgAdmin 4 interface. The left pane displays the database structure, including the 'cd' schema and the 'facilities' table. The central pane shows a SQL query: `select * from cd.facilities;`. The bottom pane displays the results of the query in a table format.

	facid	name	membercost	guestcost	initialoutlay	monthlymaintenance
1	0	Tennis Court 1	5	25	10000	200
2	1	Tennis Court 2	5	25	8000	200
3	2	Badminton Court	0	15.5	4000	50
4	3	Table Tennis	0	5	320	10
5	4	Massage Room 1	35	80	4000	3000
6	5	Massage Room 2	35	80	4000	3000
7	6	Squash Court	3.5	17.5	5000	80
8	7	Snooker Table	0	5	450	15
9	8	Pool Table	0	5	400	15

Total rows: 9 of 9 Query complete 00:00:00.107

- b. Retrieve specific columns from a table.



The screenshot shows the pgAdmin 4 interface. The left pane displays the database structure, including the 'cd' schema and the 'facilities' table. The central pane shows a SQL query: `select name, membercost from cd.facilities;`. The bottom pane displays the results of the query in a table format.

	name	membercost
1	Tennis Court 1	5
2	Tennis Court 2	5
3	Badminton Court	0
4	Table Tennis	0
5	Massage Room 1	35
6	Massage Room 2	35
7	Squash Court	3.5
8	Snooker Table	0
9	Pool Table	0

Total rows: 9 of 9 Query complete 00:00:00.097



c. Control which rows are retrieved.

Query: `select * from cd.facilities where membercost > 0;`

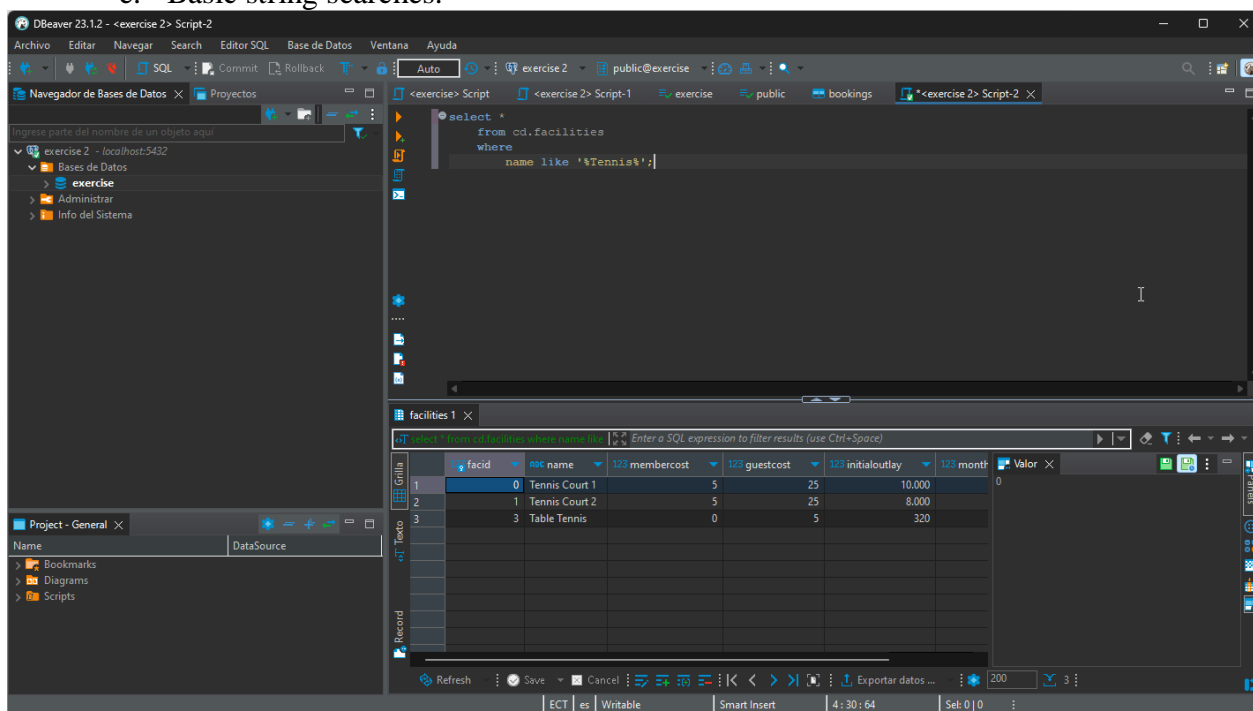
facid	name	membercost	guestcost	initialoutlay	monthlymaintenance
1	0 Tennis Court 1	5	25	10000	200
2	1 Tennis Court 2	5	25	8000	200
3	4 Massage Room 1	35	80	4000	3000
4	5 Massage Room 2	35	80	4000	3000
5	6 Squash Court	3.5	17.5	5000	80

d. Control which rows are retrieved – part 2.

Query: `select facid, name, membercost, monthlymaintenance
from cd.facilities
where
membercost > 0 and
(membercost < monthlymaintenance/50.0);`

facid	name	membercost	monthlymaintenance
1	4 Massage Room 1	35	3000
2	5 Massage Room 2	35	3000

e. Basic string searches.



3. Realice las siguientes consultas a la base de datos:

- Mostrar una lista de las instalaciones con etiquetas de “barato” si el costo del mantenimiento es menor a 200 y “caro” si es mayor.

Query Query History

```
1 SELECT name, monthlymaintenance,
2 CASE
3 WHEN monthlymaintenance >= 200 THEN 'Caro'
4 ELSE 'Barato'
5 END AS Caro
6 FROM
7 cd.facilities;
```

Data Output Messages Notifications

	name character varying (100)	monthlymaintenance numeric	Caro text
1	Tennis Court 1	200	Caro
2	Tennis Court 2	200	Caro
3	Badminton Court	50	Barato
4	Table Tennis	10	Barato
5	Massage Room 1	3000	Caro
6	Massage Room 2	3000	Caro
7	Squash Court	80	Barato
8	Snooker Table	15	Barato
9	Pool Table	15	Barato



Query

```
1 SELECT name,  
2 CASE WHEN monthlymaintenance < 200 THEN 'barato'  
3 ELSE 'caro' END AS etiqueta  
4 FROM cd.facilities;
```

Data Output Messages Notifications

	name character varying (100)	etiqueta text
1	Tennis Court 1	caro
2	Tennis Court 2	caro
3	Badminton Court	barato
4	Table Tennis	barato
5	Massage Room 1	caro
6	Massage Room 2	caro
7	Squash Court	barato
8	Snooker Table	barato
9	Pool Table	barato

- b. Mostrar en orden ascendente los 10 primeros apellidos de los miembros del club, el listado no debe contener duplicados.

Query

```
1 SELECT DISTINCT surname  
2 FROM cd.members  
3 ORDER BY surname  
4 LIMIT 10;
```

Data Output Messages Notifications

	surname character varying (200)
1	Bader
2	Baker
3	Boothe
4	Butters
5	Coplin
6	Crumpet
7	Dare
8	Farrell
9	Genting
10	GUEST



- c. Listar los starttime de las reservas de las canchas para la fecha 21-09-2012 y en otra columna el nombre de las instalaciones que corresponden a esas fechas, ordenar por tiempo ascendente.

Query Query History

```
1 SELECT b.starttime as "Reservas", f.name as "Nombre Instalaciones"
2 FROM cd.bookings b
3 JOIN cd.facilities f ON b.facid = b.facid
4 WHERE DATE(b.starttime) = '2012-09-21'
5 ORDER BY b.starttime ASC;
6
7
```

Data Output Messages Notifications

	Reservas timestamp without time zone	Nombre Instalaciones character varying (100)
1	2012-09-21 08:00:00	Table Tennis
2	2012-09-21 08:00:00	Badminton Court
3	2012-09-21 08:00:00	Tennis Court 2
4	2012-09-21 08:00:00	Tennis Court 1
5	2012-09-21 08:00:00	Tennis Court 1
6	2012-09-21 08:00:00	Pool Table
7	2012-09-21 08:00:00	Snooker Table
8	2012-09-21 08:00:00	Squash Court
9	2012-09-21 08:00:00	Massage Room 2
10	2012-09-21 08:00:00	Pool Table
11	2012-09-21 08:00:00	Massage Room 1

Total rows: 621 of 621 Query complete 00:00:00.068



- d. Listar los miembros que han utilizado la cancha de tennis, incluir el nombre de la cancha y el nombre del miembro en una sola columna.

Sin Repeticiones

Query Query History

```
1 SELECT distinct CONCAT(f.name, ' - ', m.firstname, ' ', m.surname) AS "Cancha de Tennis - Miembro"
2 FROM cd.members m
3 JOIN cd.bookings b ON m.memid = b.memid
4 JOIN cd.facilities f ON f.facid = b.facid
5 WHERE f.name LIKE '%Tennis%';
6
```

Data Output Messages Notifications

	Cancha de Tennis - Miembro text
1	Tennis Court 2 - Nancy Dare
2	Tennis Court 1 - Ponder Stibbons
3	Table Tennis - David Pinker
4	Table Tennis - Jemima Farrell
5	Table Tennis - John Hunt
6	Table Tennis - Charles Owen
7	Tennis Court 1 - Joan Coplin
8	Tennis Court 2 - David Jones
9	Tennis Court 1 - Jack Smith
10	Tennis Court 2 - Jemima Farrell
11	Table Tennis - James Testette

Total rows: 72 of 72 Query complete 00:00:00.063



Con Repeticiones

Query

```
1 SELECT CONCAT(m.firstname, ' - ', f.name) AS miembro_y_cancha
2 FROM cd.members m
3 JOIN cd.bookings r ON m.memid = r.memid
4 JOIN cd.facilities f ON f.facid=r.facid
5 WHERE f.name LIKE '%Tennis%';
```

Data Output Messages Notifications



	miembro_y_cancha text
1	GUEST - Tennis Court 1
2	GUEST - Tennis Court 1
3	GUEST - Tennis Court 1
4	GUEST - Tennis Court 1
5	GUEST - Tennis Court 1
6	GUEST - Tennis Court 1
7	GUEST - Tennis Court 1
8	GUEST - Tennis Court 1
9	GUEST - Tennis Court 1
10	GUEST - Tennis Court 1
11	GUEST - Tennis Court 1
12	GUEST - Tennis Court 1
13	GUEST - Tennis Court 1
14	GUEST - Tennis Court 1

Total rows: 1000 of 1200 Query complete 00:00:00.116



- e. Listar los miembros quienes han recomendado a otro miembro. El resultado de la consulta no debe tener duplicados y deben estar ordenados primero por el apellido y luego por el primer nombre.

Query Query History

```
1 SELECT DISTINCT surname, firstname
2 FROM cd.members
3 WHERE recommendedby != 0
4 ORDER BY surname ASC;
5
```

Data Output Messages Notifications

	surname character varying (200) 🔒	firstname character varying (200) 🔒
1	Bader	Florence
2	Baker	Anne
3	Baker	Timothy
4	Boothe	Tim
5	Butters	Gerald
6	Coplin	Joan
7	Crumpet	Erica
8	Dare	Nancy
9	Genting	Matthew
10	Hunt	John
11	Jones	David

Total rows: 22 of 22 Query complete 00:00:00.061



- f. Borrar de la base de datos al miembro cuyo id es 37 quien nunca ha realizado una reserva.

Query

```
1 DELETE FROM cd.members
2 WHERE memid = 37 AND NOT EXISTS (
3     SELECT 1
4     FROM cd.bookings
5     WHERE memid = 37
6 );
```

Data Output Messages Notifications

DELETE 1

Query returned successfully in 60 msec.

Query Query History Scratch Pad

```
1 select * from cd.members;
2 select * from cd.facilities;
3 select * from cd.bookings;
```

Data Output Messages Notifications

	memid [PK] integer	surname character varying (200)	firstname character varying (200)	address character varying (300)	zipcode integer	telephone character varying (20)	recommendedby integer	joindate timestamp without time zc
21	22	Coplin	Joan	85 Bard Street, Bloomington, Boston	43533	(822) 499-2232	16	2012-08-29 08:32:41
22	24	Sarwin	Ramnaresh	12 Bullington Lane, Boston	65464	(822) 413-1470	15	2012-09-01 08:44:42
23	26	Jones	Douglas	976 Gnats Close, Reading	11986	844 536-8036	11	2012-09-02 18:43:05
24	27	Rumney	Henrietta	3 Burkington Plaza, Boston	78533	(822) 989-8876	20	2012-09-05 08:42:35
25	28	Farrell	David	437 Granite Farm Road, Westford	43532	(855) 755-9876	[null]	2012-09-15 08:22:05
26	29	Worthington-Smyth	Henry	55 Jagbi Way, North Reading	97676	(855) 894-3758	2	2012-09-17 12:27:15
27	30	Purview	Millicent	641 Drudgery Close, Burnington, Boston	34232	(855) 941-9786	2	2012-09-18 19:04:01
28	33	Tupperware	Hyacinth	33 Cheerful Plaza, Drake Road, Westford	68666	(822) 665-5327	[null]	2012-09-18 19:32:05
29	35	Hunt	John	5 Bullington Lane, Boston	54333	(899) 720-6978	30	2012-09-19 11:32:45
30	36	Crumpet	Erica	Crimson Road, North Reading	75655	(811) 732-4816	2	2012-09-22 08:36:38

Query Scratch Pad

```
1 select * from cd.members where memid=37;
```

Data Output Messages Notifications

memid [PK] integer	surname character varying (200)	firstname character varying (200)	address character varying (300)	zipcode integer	telephone character varying (20)	recommendedby integer	joindate timestamp without time zor
-----------------------	------------------------------------	--------------------------------------	------------------------------------	--------------------	-------------------------------------	--------------------------	--

Query History
pgAdmin? affected
Remove
Remove All

RESULTADOS OBTENIDOS:

En los ejercicios que realizamos, aprendimos mucho sobre cómo trabajar con bases de datos y hacer consultas para obtener información específica. En el primer ejercicio, descargamos e instalamos una base de datos llamada "Employees" de MySQL, en la cual obtuvimos acceso a muchos datos ya organizados. Aprendimos cómo cargar información en tablas y asegurarnos de que los datos fueran válidos y realistas.

También aprendimos a manejar algunos problemas técnicos, como ajustar la capacidad de almacenamiento para evitar errores relacionados con el tamaño de los registros. A medida que avanzamos, ejecutamos pruebas para hacer diferentes consultas y ver cómo interactuaban con la base de datos.

Para el segundo ejercicio, trabajamos con la base de datos llamada "exercises" en PostgreSQL. Fue un poco más complicado, pero logramos importarla y realizar consultas más complejas. Aprendimos cómo seleccionar solo las columnas que necesitábamos y cómo filtrar la información para obtener solo lo que nos interesaba.

En general, estos ejercicios nos dieron una visión más amplia de cómo trabajar con bases de datos y cómo usar consultas SQL para obtener datos específicos. Los cuales nos permiten mejorar nuestras habilidades las cuales podrán ser muy útiles en nuestras carreras profesionales.

CONCLUSIONES:

- La práctica de laboratorio se enfocó en comparar la carga de datos y realizar consultas entre MySQL y PostgreSQL. Ambos sistemas de gestión de bases de datos demostraron ser eficientes en la carga de datos de prueba, sin presentar diferencias significativas en el tiempo de carga para conjuntos de datos similares.
- En cuanto a las consultas de búsqueda básicas, ambas bases de datos exhibieron un rendimiento similar, con tiempos de respuesta prácticamente iguales.
- Si bien no se realizaron mediciones de rendimiento en esta práctica, se pudo observar que ambas bases de datos son sólidas y eficientes para la mayoría de las tareas de bases de datos en un entorno de laboratorio.
- La elección entre MySQL y PostgreSQL dependerá de las necesidades específicas del proyecto, el tipo de consultas que se realizarán y los recursos disponibles para el despliegue y mantenimiento del sistema.

RECOMENDACIONES:

- ✓ Antes de comenzar a realizar las consultas y sus tablas resultantes se debe tener un previo conocimiento de las cláusulas principales de postgres (FROM y SELECT), ya que son los que se van a usar para dar solución a los enunciados de los ejercicios planteados.
- ✓ Para realizar las diferentes consultas acordes al enunciado de cada literal de cada ejercicio, se debe leer las veces que sean necesarias hasta entender el contexto que nos pide, es decir, revisar una y otra vez las tablas de las bases de datos y comprender de que tablas nos pide el enunciado extraer la información o los datos necesarios para construir la expresión en SQL y correctamente.



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

GUÍA PARA LAS PRÁCTICAS DE LABORATORIO, TALLER O CAMPO.

CÓDIGO: SGC.DI.505
VERSIÓN: 2.0
FECHA ÚLTIMA
REVISIÓN: 12/04/2017

- ✓ Para realizar las tablas resultantes de las consultas mediante SQL es conveniente empezar analizando desde las expresiones más pequeñas hasta las expresiones más grandes, de esta manera se consigue dibujar las tablas resultantes sin tantas complicaciones.

ELABORADO POR:

Rivadeneira Gómez Ricardo Xavier
Medina Armijos Martin Andrés
Yanacallo Monta Angelo Josué
Imbaquinga Guaña Jose Ricardo

ESTUDIANTE