

# 1:

## CONFIG AWS:

The screenshot shows the AWS Academy Learner Lab interface. On the left is a dark sidebar with navigation icons for Usuario, Panel de control, Asignaturas, Calendario, Bandeja de entrada, Historial, and Ayuda. The main content area has a top header with the breadcrumb 'ALLv2ES-E... > Contenidos > Laboratori...' and a sub-header 'Lanzamiento del Laboratorio para el alumnado de AWS Academy'. Below this is a status bar showing 'AWS' with a yellow dot, 'Used \$3.6 of \$50', and a timer '00:00'. The main workspace is divided into two sections: a large central area with a blue 'V' logo and a right-hand sidebar titled 'Learner Lab' containing a list of links like 'Environment Overview', 'Access the AWS Management Console', etc. At the bottom of the workspace are 'Anterior' and 'Siguiente' buttons.

The screenshot shows the AWS Management Console search results for 'rds'. The search bar at the top contains 'rds'. Below it, the results are categorized under 'Servicios (11)'. The first result is 'RDS', described as 'Servicio de bases de datos relacionales administrado'. It includes a section for 'Características principales' with links to 'Panel', 'Bases de datos', 'Editor de consultas', and 'Información sobre rendimiento'. The second result is 'AWS Glue DataBrew', described as 'Herramienta de preparación de datos visuales para limpiar y normalizar datos para ...'. The interface is dark-themed with a sidebar on the left showing various resource counts like 'Características (15)', 'Recursos (New)', etc.

aws

Servicios

Q Buscar

[Alt+S]

RDS > Crear base de datos

Información

Crear base de datos

Información


Elegir un método de creación de base de datos


☐ Creación estándar  
Puede definir todas las opciones de configuración, incluidas las de disponibilidad, seguridad, copias de seguridad y mantenimiento.


☒ Creación sencilla  
Utilice las configuraciones recomendadas. Algunas opciones de configuración se pueden cambiar después de crear la base de datos.


Configuración


Tipo de motor Información


☐ Aurora (MySQL Compatible)  



☐ Aurora (PostgreSQL Compatible)  


☒ MySQL  


☐ MariaDB  


☐ PostgreSQL  


☐ Oracle  


☐ Microsoft SQL Server  


aws

Servicios

Buscar

[Alt+S]

Comunidad de MySQL

Tamaño de la instancia de base de datos

☐ Producción

db.r6g.xlarge

4 vCPUs

32 GiB RAM

500 GiB

☐ Desarrollo y pruebas

db.r6g.large

2 vCPUs

16 GiB RAM

100 GiB

☒ Capa gratuita

db.t4g.micro

2 vCPUs

1 GiB RAM

20 GiB

Identificador de instancias de bases de datos

Escriba un nombre para la instancia de base de datos. El nombre debe ser único en relación con todas las instancias de base de datos pertenecientes a su cuenta de AWS en la región de AWS actual.

riomaio-re-joel

El identificador de la instancia de base de datos no distingue entre mayúsculas y minúsculas, pero se almacena con todas las letras en minúsculas (como en "miinstanciadebd"). Restricciones: de 1 a 60 caracteres alfanuméricos o guiones. El primer carácter debe ser una letra. No puede contener dos guiones consecutivos. No puede terminar con un guion.

Nombre de usuario maestro

Información

Escriba un ID de inicio de sesión para el usuario maestro de la instancia de base de datos.

admin

1 a 16 caracteres alfanuméricos. El primer carácter debe ser una letra.

Administración de credenciales

Puede usar AWS Secrets Manager o administrar sus credenciales de usuario maestro.

☐ Administrado en AWS Secrets Manager - más seguro

RDS genera una contraseña y la administra durante todo su ciclo de vida mediante AWS Secrets Manager.

☒ Autoadministrado

Cree su propia contraseña o pida a RDS que cree una contraseña para que pueda administrarla.

☐ Generar contraseña automáticamente

Amazon RDS puede generar una contraseña en su nombre, o bien puede especificar su propia contraseña.

Contraseña maestra

Información

\*\*\*\*\*

Password strength

Very weak

Restricciones mínimas: al menos 8 caracteres ASCII imprimibles. No puede contener ninguno de los siguientes símbolos: / \* " @

Se ha creado correctamente la base de datos riomaio-re-joel

Ver detalles de conexión

Puede utilizar la configuración de riomaio-re-joel para simplificar la configuración de complementos de base de datos sugeridos mientras terminamos de crear su base de datos.

Presentación de los puntos de conexión globales

Cada clúster global tiene un punto de conexión global que la aplicación puede usar para enviar solicitudes a la instancia de escritura del clúster principal. Tras una operación de conmutación por error o conmutación, este punto de conexión envía automáticamente las solicitudes a la instancia de escritura del nuevo clúster principal para que no tenga que cambiar el código de la aplicación. Más información

Notificaciones 0 0 0 0 2 0

RDS > Bases de datos

Bases de datos (3)

Recursos del grupo

Modificar

Acciones

Restaurar desde S3

Crear base de datos

Filtrar por bases de datos

	Identificador de base de datos	Estado	Rol	Motor	Región y...	Tamaño	Recomendaciones	CPU	Activida...	Manteni...
<input type="radio"/>	database-2	Disponible	Instancia	MySQL Co...	us-east-1f	db.t4g.micro	2 Informativo	3.55%	0 Conexi	
<input type="radio"/>	repaso	Disponible	Instancia	MySQL Co...	us-east-1b	db.t4g.micro	2 Informativo	2.88%	0 Conexi	
<input type="radio"/>	riomaio-re-joel	Disponible	Instancia	MySQL Co...	us-east-1a	db.t4g.micro		-		

RDS > Bases de datos > riomaio-re-joel

riomaio-re-joel

Modificar

Servicios

Buscar

[Alt+S]

Grupo de seguridad

Lista de grupos de seguridad de base de datos que se asociarán a esta instancia de base de datos.

Elegir grupos de seguridad

default

Entidad de certificación

Información

Al utilizar un certificado de servidor, se obtiene una capa adicional de seguridad al validar que la conexión se establece con una base de datos de Amazon. Para ello, se comprueba el certificado de servidor que se instala automáticamente en todas las bases de datos aprovisionadas.

rds-ca-rsa2048-g1 (predeterminado)

Vencimiento: May 26, 2061

Configuración adicional

Acceso público

Accesible públicamente

RDS asigna una dirección IP pública a la base de datos. Las instancias de Amazon EC2 y otros recursos fuera de la VPC pueden conectarse a la base de datos. Los recursos de la VPC también pueden conectarse a la base de datos. Elija uno o varios grupos de seguridad de VPC que especifiquen qué recursos pueden conectarse a la base de datos.

No accesible públicamente

No se asigna ninguna dirección IP a la instancia de base de datos. Las instancias EC2 y los dispositivos fuera de la VPC no pueden conectarse.

Puerto de la base de datos

Especifique el puerto TCP/IP que la instancia de base de datos utilizará para las conexiones de la aplicación. La cadena de conexión de la aplicación debe especificar el número de puerto. El grupo de seguridad de base de datos y el firewall deben permitir conexiones al puerto.

Más información

3306

Conectividad

Tipo de red

Información

Para utilizar el modo de pila doble, asegúrese de asociar un bloque de CIDR IPv6 a una subred en la VPC que especifique.

IPv4

Sus recursos solo pueden comunicarse a través del protocolo de direcciones IPv4.

Modo de pila doble

Sus recursos pueden comunicarse a través de IPv4, IPv6 o ambos.

Grupo de subredes de la base de datos

default-vpc-03903172208765c33

Grupo de seguridad

Lista de grupos de seguridad de base de datos que se asociarán a esta instancia de base de datos.

Elegir grupos de seguridad

AccesoDesdeFuera

Entidad de certificación

Información

Al utilizar un certificado de servidor, se obtiene una capa adicional de seguridad al validar que la conexión se establece con una base de datos de Amazon. Para ello, se comprueba el certificado de servidor que se instala automáticamente en todas las bases de datos aprovisionadas.

rds-ca-rsa2048-g1 (predeterminado)

Vencimiento: May 26, 2061

AccesoDesdeFuera es el que creamos en clase , por lo que tiene la misma configuracion que se hizo el día de clase.

## Modificar la instancia de base de datos: riomaioire-joel

### Resumen de modificaciones

Está a punto de enviar las siguientes modificaciones. Solo se muestran los valores que van a modificarse. Revise con cuidado los cambios y haga clic en Modificar la instancia de base de datos.

Atributo	Valor actual	Valor nuevo
Grupo de seguridad	default	AccesoDesdeFuera
Acceso público	No	Sí

### Programación de modificaciones

Cuándo aplicar las modificaciones

☐ Aplicar durante el próximo periodo de mantenimiento programado

Periodo de mantenimiento actual: October 28, 2024 11:29 - 11:59 (UTC+02:00)

☒ Aplicar inmediatamente

Los cambios en esta solicitud y cualquier modificación pendiente se aplicarán de forma asíncrona lo antes posible, independientemente de la configuración del periodo de mantenimiento para esta instancia de base de datos.

Cancelar

Atrás

Modificar la instancia de base de datos

para obtener la captura de la consulta, primero hacemos el siguiente paso para estar conectado mediante un cliente y poder hacer la consulta pero el resultado seria este:

The screenshot shows the MySQL Workbench interface. The 'Query Editor' window contains the following SQL query:

```
1 • SELECT create_time FROM INFORMATION_SCHEMA.TABLES;  
2
```

The 'Result Grid' window displays the execution results of the query. The results are as follows:

CREATE_TIME
2024-10-24 05:55:42
2024-10-24 05:55:42
2024-10-24 05:55:44
2024-10-24 05:55:44
2024-10-24 05:55:44
2024-10-24 05:55:44
2024-10-24 05:55:44
2024-10-24 05:55:44

The 'Output' window at the bottom shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	08:11:42	SELECT create_time FROM INFORMATION_SCHEMA.TABLES LIMIT 0, ...	339 row(s) returned	0.109 sec / 0.000 sec

## 2: CONECTARSE MEDIANTE CLIENTE

×

# Welcome to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

[Browse Documentation >](#)[Read the Blog >](#)[Discuss on the Forums >](#)

### MySQL Connections

Filter connections

Setup New Connection

Connection Name: ExamenAdat

Type a name for the connection

Connection Method: Standard (TCP/IP)

Method to use to connect to the RDBMS

Parameters

SSL

Advanced

Hostname: :y2mujwe.us-east-1.rds.amazonaws.com

Port: 3306

Name or IP address of the server host - and TCP/IP port.

Username: admin

Name of the user to connect with.

Password: 

Store in Vault ...

Clear


The user's password. Will be requested later if it's not set.

Default Schema:

The schema to use as default schema. Leave blank to select it later.

Store Password For Connection

Please enter password for the following service:



Service: Mysql@riomaio-re-joel.cjmqcy2mujwe.us-east-1.rds.amazonaws.com:3306

User: admin

Password: 

\*\*\*\*\*

OK

Cancel

Configure Server Management...

Test Connection


Cancel


OK

hostname es el enlace del AWS,

## Conectividad y seguridad

### Punto de enlace y puerto

 Punto de enlace copiado

 riomaiores-joel.cjqmcy2mujwe.us-east-1.rds.amazonaws.com

Puerto

3306

Setup New Connection

Connection Name: ExamenAdat Type a name for the connection

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters

SSL

Advanced

Hostname: y2mujwe.us-east-1.rds.amazonaws.com

Port: 3306

Name or IP address of the server host - and TCP/IP port.

Username: admin

Name of the user to connect with.

Password: 

Store in Vault ...

Clear

The user's password. Will be requested later if it's not set.

Default Schema:


The schema to use as default schema. Leave blank to select it later.

Test Connection

Cancel

OK

MySQL Workbench

 Successfully made the MySQL connection

Information related to this connection:

Host:  
riomaiores-joel.cjqmcy2mujwe.us-east-1.rds.amazonaws....  
Port: 3306  
User: admin  
SSL: enabled with TLS\_AES\_128\_GCM\_SHA256

A successful MySQL connection was made with the parameters defined for this connection.

OK

### 3. CONECTARSE MEDIANTE JDBC

para ello, he creado un archivo JAVA llamado ConexionBBDD.java que es en el que contengo el código de esta actividad. Las líneas clave que modifique de un conector normal son las siguientes:

```
private static final String DB_URL =  
"jdbc:mysql://riomaiore-joel.cjqmcy2mujwe.us-east-1.rds.amazonaws.com:3  
306/";  
    private static final String USER = "admin";  
    private static final String PASSWORD = "12345678";
```

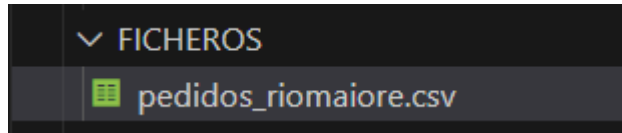
y el output muestra que se ha realizado correctamente:

```
PS C:\Users\HP\Desktop\DM2\ADAT\EXAMENES\T3\EXAMENJoe1\JavaProyectExamen>  
tb7moq5xmcdtjob.argfile' 'conexionBBDD.ConexionJDBC_AWS'  
Driver cargado correctamente.  
Conexión establecida con éxito!  
Conexión cerrada.  
PS C:\Users\HP\Desktop\DM2\ADAT\EXAMENES\T3\EXAMENJoe1\JavaProyectExamen>
```

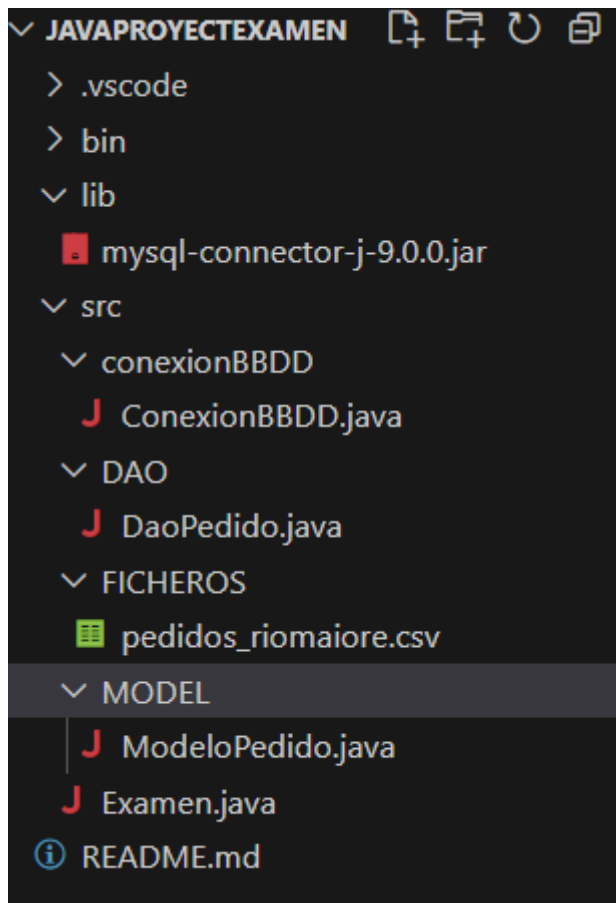


## 4. CARGAR DATOS EN EL AWS

he metido el CSV en una carpeta dentro del javaproject llamada ficheros:



para hacer la carga, he dealizado la siguiente estructura de directorios (y es la que usare durante todo el examen)



primero tenemos la clase conexion DDBB que ahora tiene el contenido para conectarse a la BBDD de AWS. en la clase examen se hace la llamada a DaoPedido, que primero, crea la tabla si no estaba creada (y yo puse que la eliminara para volver a crearla cada vez que se ejecute ya que es una carga diminuta de datos) y despues, tiene un metodo para cargar los datos leidos por el csv y que los convierte a Objetos ModeloPedido. que es una clase que recibe datos para todos los atributo que hay en el csv. EL CODIGO SE APORTA CON LA ENTREGA.

una vez ejecutado, en el workbench le damos a refresh all y sale esto al darle a ver las filas:

MySQL Workbench

ExamenAdat

FileEditViewQueryDatabaseServerToolsScriptingHelp

Navigator

SCHEMAS

Filter objects

pedidos

Tables

pedidos

Views

Stored Procedures

Functions

sys

Tables

sys\_config

Views

Stored Procedures

Functions

Administration

Schemas

Information

Table: pedidos

Columns:

ID

decimal

PK

nombre\_cliente

varchar

precio

double

descripcion\_pizza

varchar

punto\_recogida

varchar

Query 1

pedidos

Limit to 1000 rows

1 • SELECT \* FROM pedidos.pedidos;

Result Grid

	ID	nombre_cliente	precio	descripcion_pizza	punto_recogida	fecha	hora
▶	1	Carlos	9.75	Hawaiana	Local 1	20241015	1250
	2	Juan	8.99	4 Quesos	Local 1	20241012	849
	3	Ana	10.25	Vegetal	Local 3	20240929	252
	4	Pedro	9.75	Diavola	Local 1	20240922	1917
	5	Carlos	10.25	Carbonara	Local 2	20240929	1231
	6	Lucia	9.75	Carbonara	Local 1	20240923	1533
	7	Laura	12.5	Vegetal	Local 1	20241004	1000
	8	Pedro	13.5	Margarita	Local 1	20241016	518
	9	Juan	9.75	Hawaiana	Local 1	20241005	501
	10	Sofia	8.99	Barbacoa	Local 2	20240930	1704
	11	Luis	13.5	Carbonara	Local 3	20241004	1909
	12	Carlos	9.75	Margarita	Local 3	20240922	2023
	13	Miguel	11.2	Carbonara	Local 3	20241002	1933
	14	Lucia	8.99	Carbonara	Local 2	20240921	2143
	15	Juan	10.25	Diavola	Local 3	20240926	958
	16	Laura	13.5	Barbacoa	Local 2	20240927	505
	17	Carlos	11.2	Barbacoa	Local 2	20240925	33
	18	Ana	13.5	Vegetal	Local 2	20240929	458
	19	Ana	10.25	4 Quesos	Local 3	20241009	124
	20	Lucia	12.5	Diavola	Local 3	20241009	117
	21	Luis	8.99	Diavola	Local 1	20241016	704
	22	Ana	13.5	Barbacoa	Local 1	20241011	2230
	23	Lucia	15	Vegetal	Local 3	20240922	1046
	24	Maria	13.5	Vegetal	Local 3	20240920	1223
	25	Sofia	13.5	Margarita	Local 1	20240925	1537
	26	Lucia	13.5	Pepperoni	Local 1	20241016	242

pedidos1 x

ApplyRevert

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

Form Editor

Field Types

Query Stats

Execution Plan

Context HelpSnippets

Output

## 5. SELECCIONAR LOS PEDIDOS DE UNA FECHA Y PUNTO DE RECOGIDA CONCRETOS

para ello, añadi un nuevo metodo en el Dao: selectPedidos, vease en el codigo.

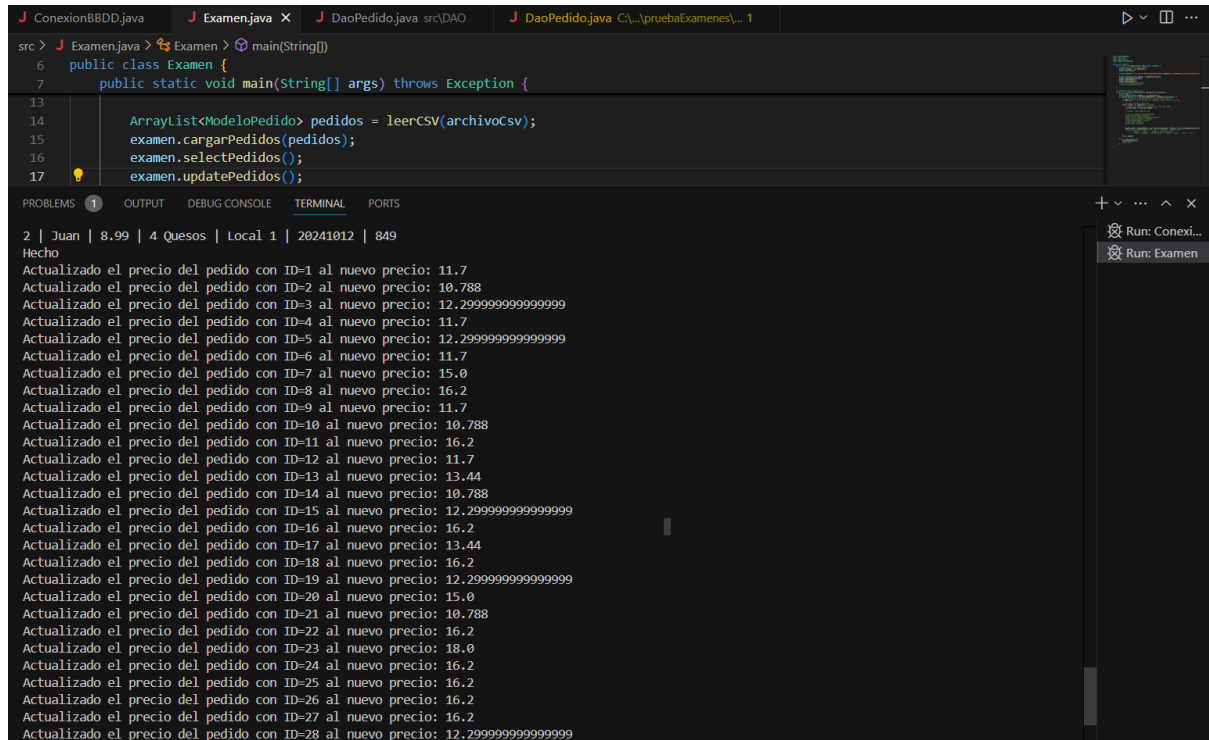
```
14      ArrayList<ModeloPedido> pedidos = leerCSV(archivoCsv);
15      examen.cargarPedidos(pedidos);
16      examen.selectPedidos();
17      //examen.actualizarPedidos();

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

... 11 more
Exception in thread "main" java.lang.NullPointerException: Cannot invoke "java.sql.Connection.createStatement()" because "conn" is null
    at DAO.DaoPedido.crearTablas(DaoPedido.java:28)
    at Examen.main(Examen.java:10)
PS C:\Users\HP\Desktop\DM2\ADAT\EXAMENES\T3\EXAMENJoel\JavaProyectExamen> ^C
PS C:\Users\HP\Desktop\DM2\ADAT\EXAMENES\T3\EXAMENJoel\JavaProyectExamen>
PS C:\Users\HP\Desktop\DM2\ADAT\EXAMENES\T3\EXAMENJoel\JavaProyectExamen> c::; cd 'c:\Users\HP\Desktop\DM2\ADAT\EXAMENES\T3\EXAMENJoel\JavaProyectExam
en'; & 'C:\Program Files\Java\jdk-23\bin\java.exe' '@C:\Users\HP\AppData\Local\Temp\cp_256xbgzp0btb7mq5xmcdtjob.job.argfile' 'Examen'
DELETED.
Tabla 'PEDIDOS' creada correctamente.
Pedidos cargados correctamente.
2 | Juan | 8.99 | 4 Quesos | Local 1 | 20241012 | 849
Hecho
PS C:\Users\HP\Desktop\DM2\ADAT\EXAMENES\T3\EXAMENJoel\JavaProyectExamen> █
```

## 6. MODIFICAR PRECIO DE TODOS LOS PEDIDOS Y AUMENTARLO EN 20%

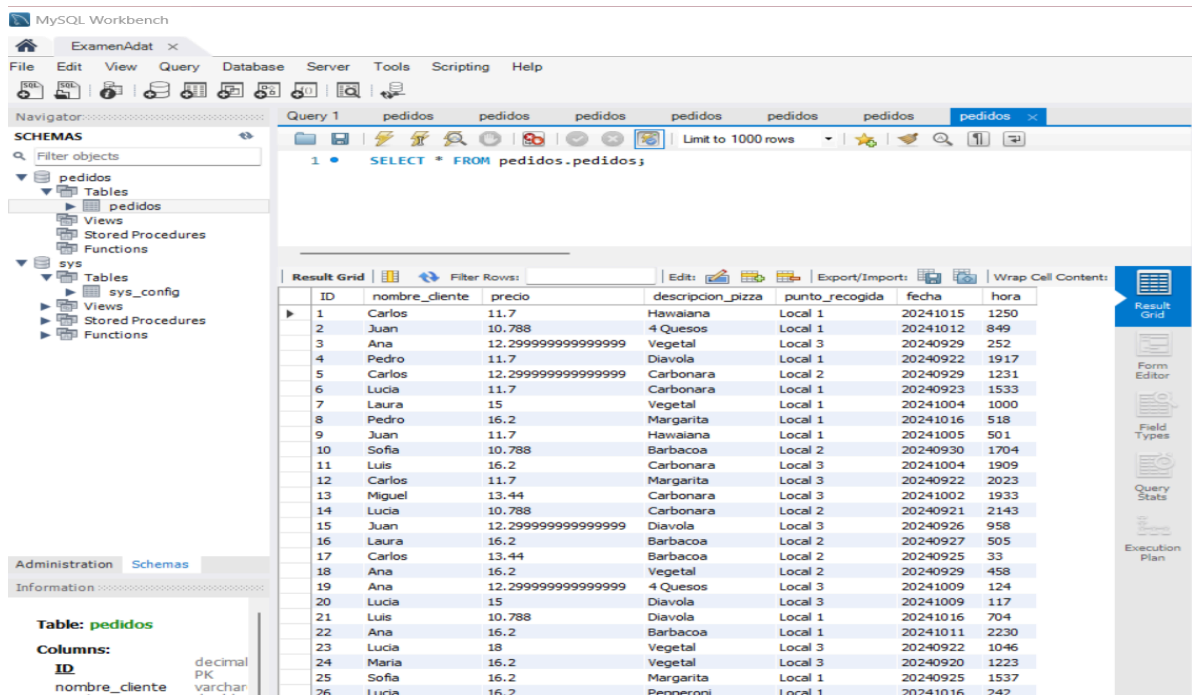
resultado:



The screenshot shows an IDE with a Java file named `Examen.java`. The code defines a `main` method that reads a CSV file, loads it into an `ArrayList`, and then calls `examen.updatePedidos()`. The output window shows the execution results, indicating that the price of all orders was updated by 20%.

```
src > J Examen.java > Examen > main(String[])
6 public class Examen {
7     public static void main(String[] args) throws Exception {
13
14         ArrayList<ModeloPedido> pedidos = leerCSV(archivoCsv);
15         examen.cargarPedidos(pedidos);
16         examen.selectPedidos();
17         examen.updatePedidos();
18     }
19 }
```

2 | Juan | 8.99 | 4 Quesos | Local 1 | 20241012 | 849  
Hecho  
Actualizado el precio del pedido con ID=1 al nuevo precio: 11.7  
Actualizado el precio del pedido con ID=2 al nuevo precio: 10.788  
Actualizado el precio del pedido con ID=3 al nuevo precio: 12.299999999999999  
Actualizado el precio del pedido con ID=4 al nuevo precio: 11.7  
Actualizado el precio del pedido con ID=5 al nuevo precio: 12.299999999999999  
Actualizado el precio del pedido con ID=6 al nuevo precio: 11.7  
Actualizado el precio del pedido con ID=7 al nuevo precio: 15.0  
Actualizado el precio del pedido con ID=8 al nuevo precio: 16.2  
Actualizado el precio del pedido con ID=9 al nuevo precio: 11.7  
Actualizado el precio del pedido con ID=10 al nuevo precio: 10.788  
Actualizado el precio del pedido con ID=11 al nuevo precio: 16.2  
Actualizado el precio del pedido con ID=12 al nuevo precio: 11.7  
Actualizado el precio del pedido con ID=13 al nuevo precio: 13.44  
Actualizado el precio del pedido con ID=14 al nuevo precio: 10.788  
Actualizado el precio del pedido con ID=15 al nuevo precio: 12.299999999999999  
Actualizado el precio del pedido con ID=16 al nuevo precio: 16.2  
Actualizado el precio del pedido con ID=17 al nuevo precio: 13.44  
Actualizado el precio del pedido con ID=18 al nuevo precio: 16.2  
Actualizado el precio del pedido con ID=19 al nuevo precio: 12.299999999999999  
Actualizado el precio del pedido con ID=20 al nuevo precio: 15.0  
Actualizado el precio del pedido con ID=21 al nuevo precio: 10.788  
Actualizado el precio del pedido con ID=22 al nuevo precio: 16.2  
Actualizado el precio del pedido con ID=23 al nuevo precio: 18.0  
Actualizado el precio del pedido con ID=24 al nuevo precio: 16.2  
Actualizado el precio del pedido con ID=25 al nuevo precio: 16.2  
Actualizado el precio del pedido con ID=26 al nuevo precio: 16.2  
Actualizado el precio del pedido con ID=27 al nuevo precio: 16.2  
Actualizado el precio del pedido con ID=28 al nuevo precio: 12.299999999999999



The screenshot shows the MySQL Workbench interface. The 'pedidos' table is selected in the Navigator. The 'Query 1' window shows the SQL query `SELECT * FROM pedidos.pedidos;`. The 'Result Grid' displays the data from the 'pedidos' table, including columns for ID, nombre\_cliente, precio, descripcion\_pizza, punto\_recogida, fecha, and hora.

ID	nombre_cliente	precio	descripcion_pizza	punto_recogida	fecha	hora
1	Carlos	11.7	Hawaiana	Local 1	20241015	1250
2	Juan	10.788	4 Quesos	Local 1	20241012	849
3	Ana	12.299999999999999	Vegetal	Local 3	20240929	252
4	Pedro	11.7	Diavola	Local 1	20240922	1917
5	Carlos	12.299999999999999	Carbonara	Local 2	20240929	1231
6	Lucia	11.7	Carbonara	Local 1	20240923	1533
7	Laura	15	Vegetal	Local 1	20241004	1000
8	Pedro	16.2	Margarita	Local 1	20241016	518
9	Juan	11.7	Hawaiana	Local 1	20241005	501
10	Sofia	10.788	Barbacoa	Local 2	20240930	1704
11	Luis	16.2	Carbonara	Local 3	20241004	1909
12	Carlos	11.7	Margarita	Local 3	20240922	2023
13	Miguel	13.44	Carbonara	Local 3	20241002	1933
14	Lucia	10.788	Carbonara	Local 2	20240921	2143
15	Juan	12.299999999999999	Diavola	Local 3	20240926	958
16	Laura	16.2	Barbacoa	Local 2	20240927	505
17	Carlos	13.44	Barbacoa	Local 2	20240925	33
18	Ana	16.2	Vegetal	Local 2	20240929	458
19	Ana	12.299999999999999	4 Quesos	Local 3	20241009	124
20	Lucia	15	Diavola	Local 3	20241009	117
21	Luis	10.788	Diavola	Local 1	20241016	704
22	Ana	16.2	Barbacoa	Local 1	20241011	2230
23	Lucia	18	Vegetal	Local 3	20240922	1046
24	Maria	16.2	Vegetal	Local 3	20240920	1223
25	Sofia	16.2	Margarita	Local 1	20240925	1537
26	Lucia	16.2	Pepperoni	Local 1	20241016	242

## 7. ELIMINAR TODOS LOS PEDIDOS DE UN DIA EN CONCRETO

resultado: (borra el id 4, 12 y 23 que corresponden al día eliminado)

The screenshot shows the MySQL Workbench interface. The Query Editor contains the following SQL query:

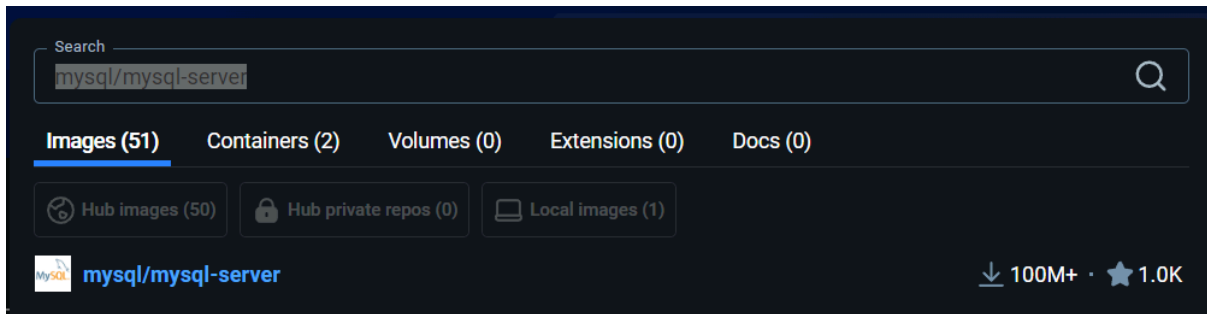
```
1 • SELECT * FROM pedidos.pedidos;
```

The Result Grid displays 29 rows of data. The columns are: ID, nombre\_cliente, precio, descripcion\_pizza, punto\_recogida, fecha, and hora. The data is as follows:

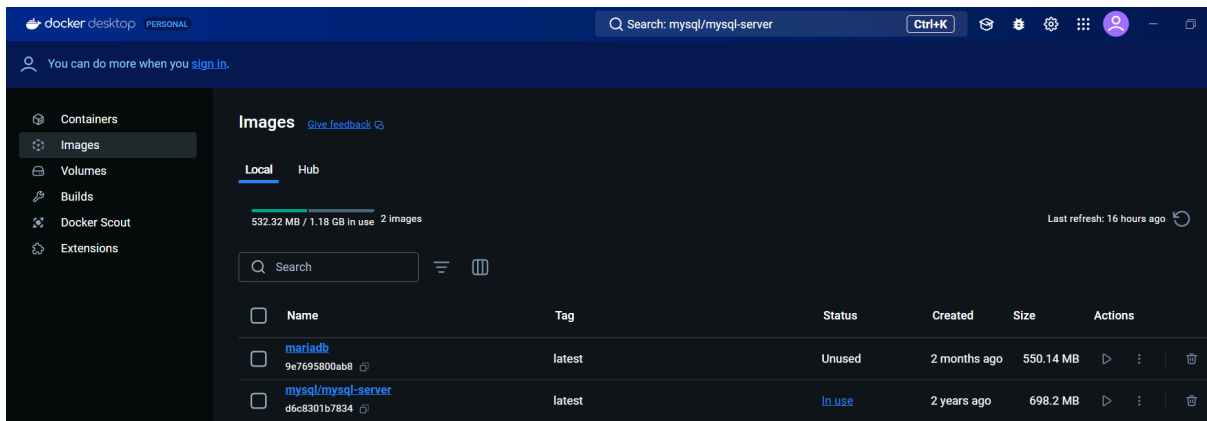
ID	nombre_cliente	precio	descripcion_pizza	punto_recogida	fecha	hora
1	Carlos	9.75	Hawaiana	Local 1	20241015	1250
2	Juan	8.99	4 Quesos	Local 1	20241012	849
3	Ana	10.25	Vegetal	Local 3	20240929	252
5	Carlos	10.25	Carbonara	Local 2	20240929	1231
6	Lucia	9.75	Carbonara	Local 1	20240923	1533
7	Laura	12.5	Vegetal	Local 1	20241004	1000
8	Pedro	13.5	Margarita	Local 1	20241016	518
9	Juan	9.75	Hawaiana	Local 1	20241005	501
10	Sofia	8.99	Barbacoa	Local 2	20240930	1704
11	Luis	13.5	Carbonara	Local 3	20241004	1909
13	Miguel	11.2	Carbonara	Local 3	20241002	1933
14	Lucia	8.99	Carbonara	Local 2	20240921	2143
15	Juan	10.25	Diavola	Local 3	20240926	958
16	Laura	13.5	Barbacoa	Local 2	20240927	505
17	Carlos	11.2	Barbacoa	Local 2	20240925	33
18	Ana	13.5	Vegetal	Local 2	20240929	458
19	Ana	10.25	4 Quesos	Local 3	20241009	124
20	Lucia	12.5	Diavola	Local 3	20241009	117
21	Luis	8.99	Diavola	Local 1	20241016	704
22	Ana	13.5	Barbacoa	Local 1	20241011	2230
24	Maria	13.5	Vegetal	Local 3	20240920	1223
25	Sofia	13.5	Margarita	Local 1	20240925	1537
26	Lucia	13.5	Pepperoni	Local 1	20241016	242
27	Pedro	13.5	Margarita	Local 3	20240924	1659
28	Juan	10.25	Barbacoa	Local 3	20241009	1329
29	Pedro	11.2	Diavola	Local 1	20240930	1136

The sidebar on the left shows the Schemas pane with the 'pedidos' database selected. The Information pane shows the table 'pedidos' with columns: ID (decimal PK), nombre\_cliente (varchar), precio (double), descripcion\_pizza (varchar), and punto\_recogida (varchar). The right sidebar contains a message about automatic context help being disabled.

## 8. CREAR CONTENEDOR DOCKER



hacemos pull, sale esto:



y ahi le damos a run en la que nos acabamos de bajar



## Run a new container

mysql/mysql-server:latest

### Optional settings



Container name

EXAMENJOEL

A random name is generated if you do not provide one.

### Ports

Enter "0" to assign randomly generated host ports.

Host port

33066

:3306/tcp

Host port

:33060/tcp

Host port

:33061/tcp

### Volumes

Host path



Container path



### Environment variables

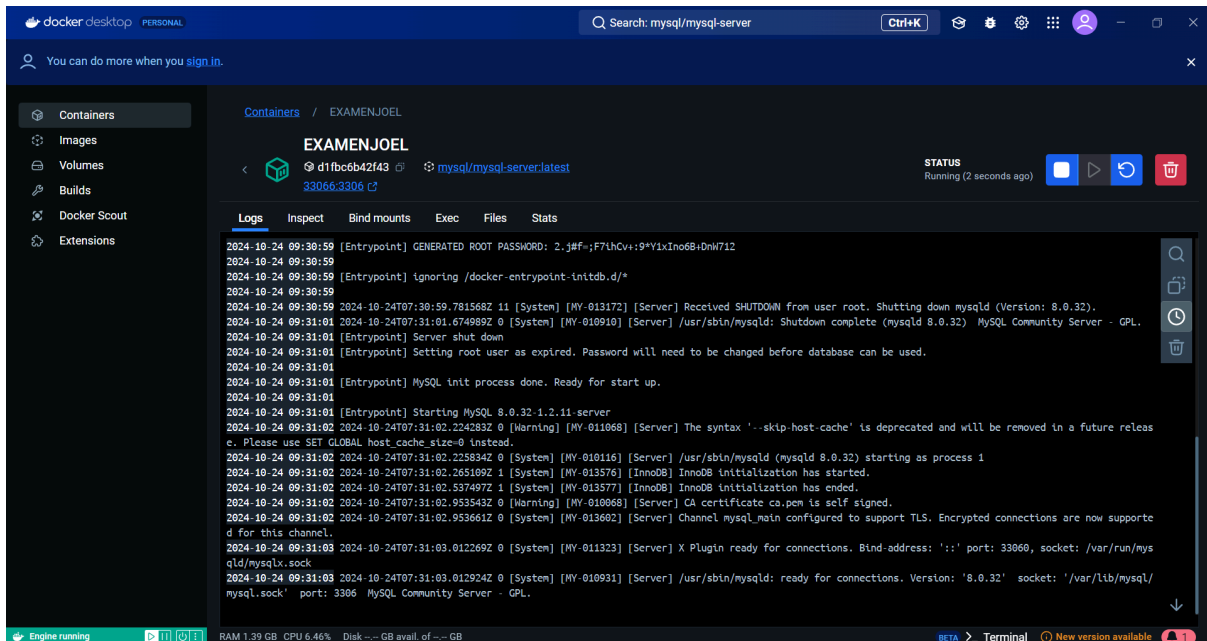
Variable

Value



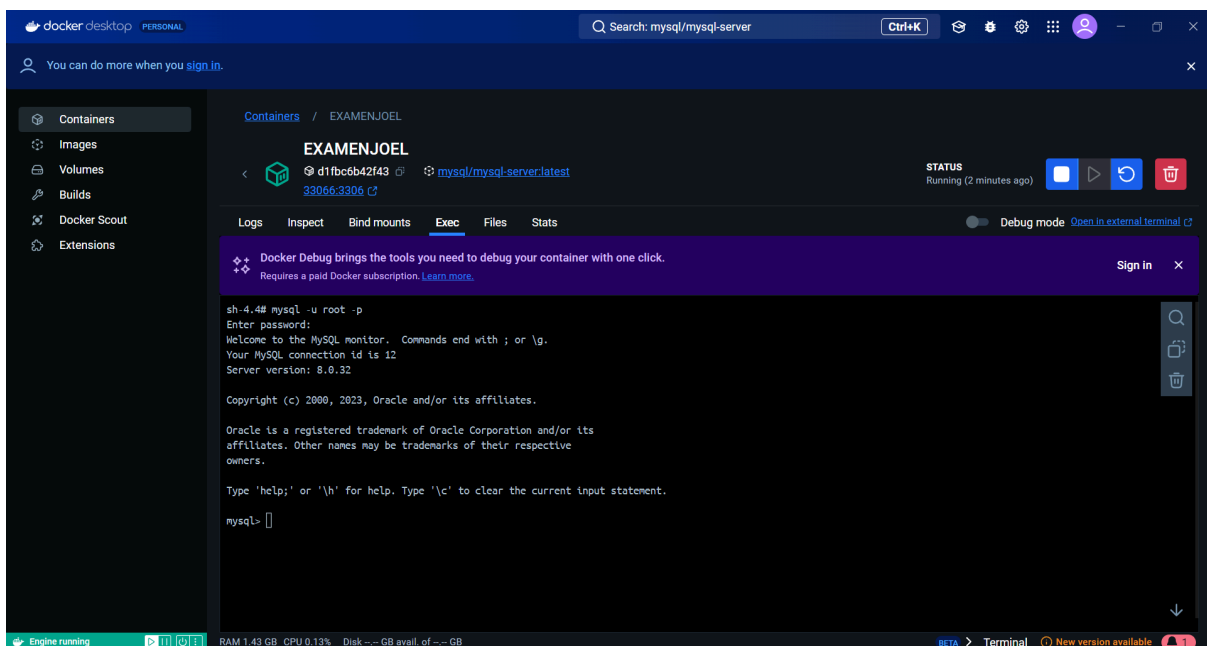
Cancel

Run



NOS COPIAMOS LA PRIMERA LINEA QUE NOS SALE EN EL LOGS, LA DE EL ROOT  
PASSWORD: 2.j#f=;F7ihCv+:9\*Y1xIno6B+DnW712

ahora vamos al exec y ejecutamos `mysql -u root -p` e introducimos esa contraseña:



ahora escribiremos los siguientes comandos:

```
# alter user 'root'@'localhost' identified by '1234';  
  
# create user 'joel'@'%' identified by 'examen';  
  
# grant all privileges on *.* to 'joel'@'%' with grant option;
```



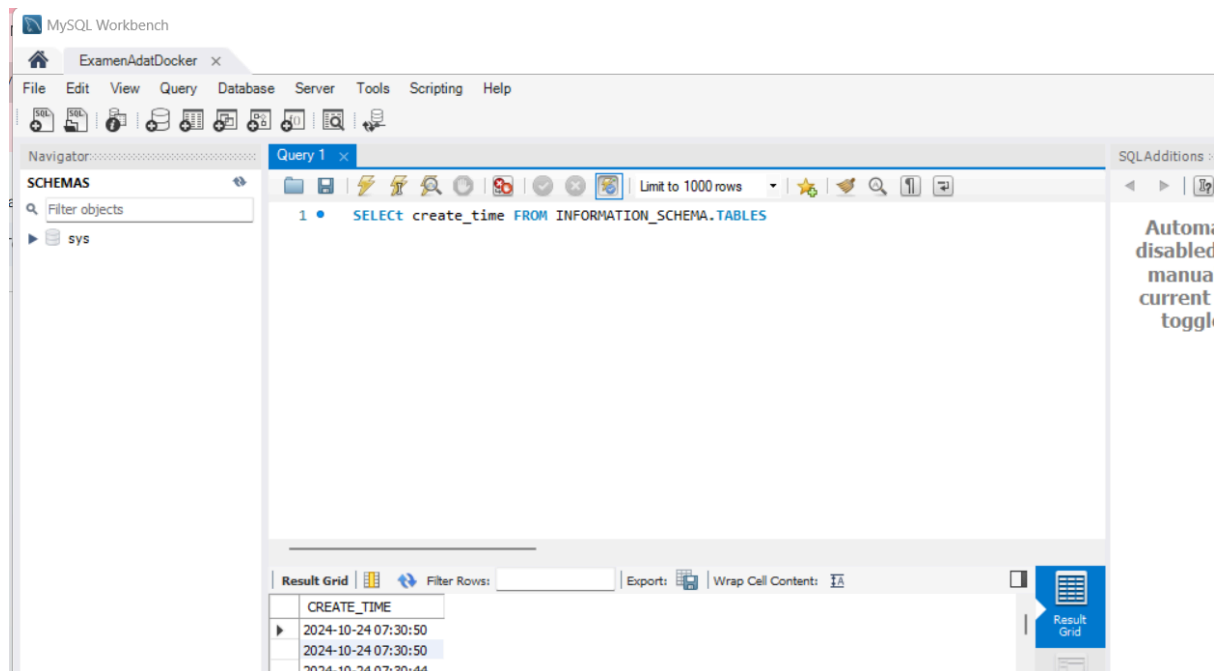
```
mysql> alter user 'root'@'localhost' identified by '1234';
Query OK, 0 rows affected (0.03 sec)

mysql> create user 'joel'@'%' identified by 'examen';
Query OK, 0 rows affected (0.05 sec)

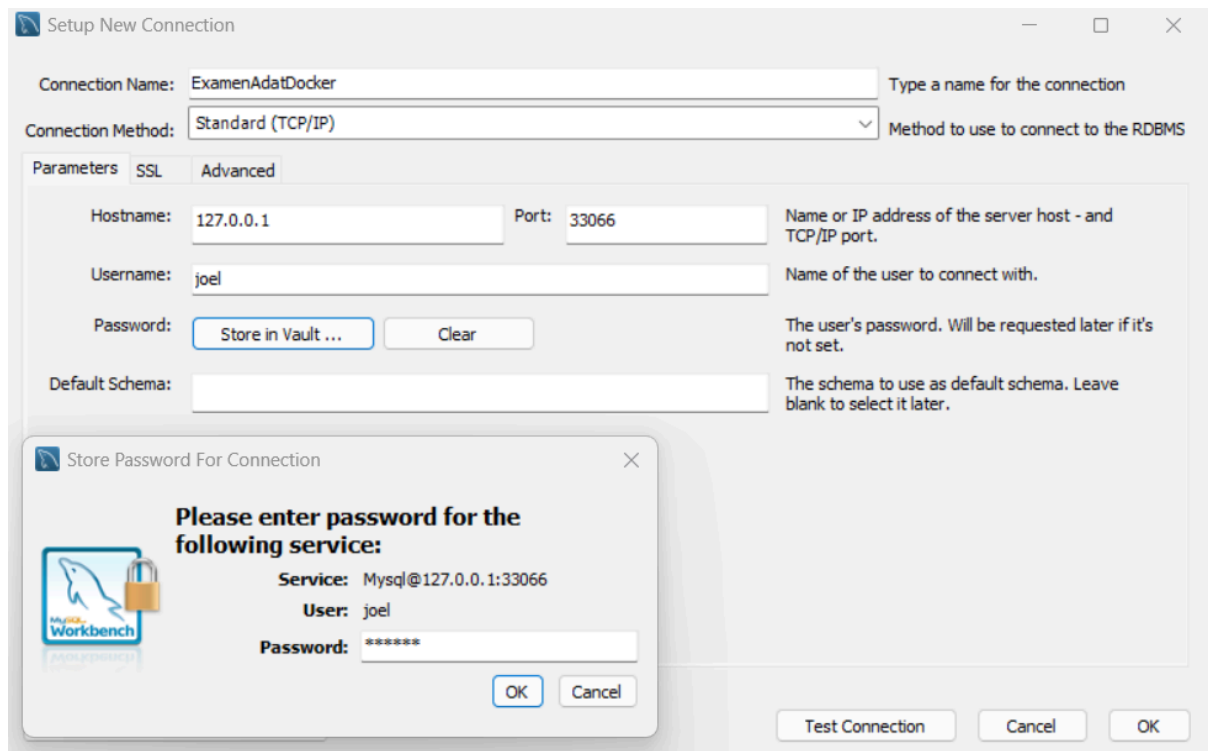
mysql> grant all privileges on *.* to 'joel'@'%' with grant option;
Query OK, 0 rows affected (0.03 sec)

mysql> 
```

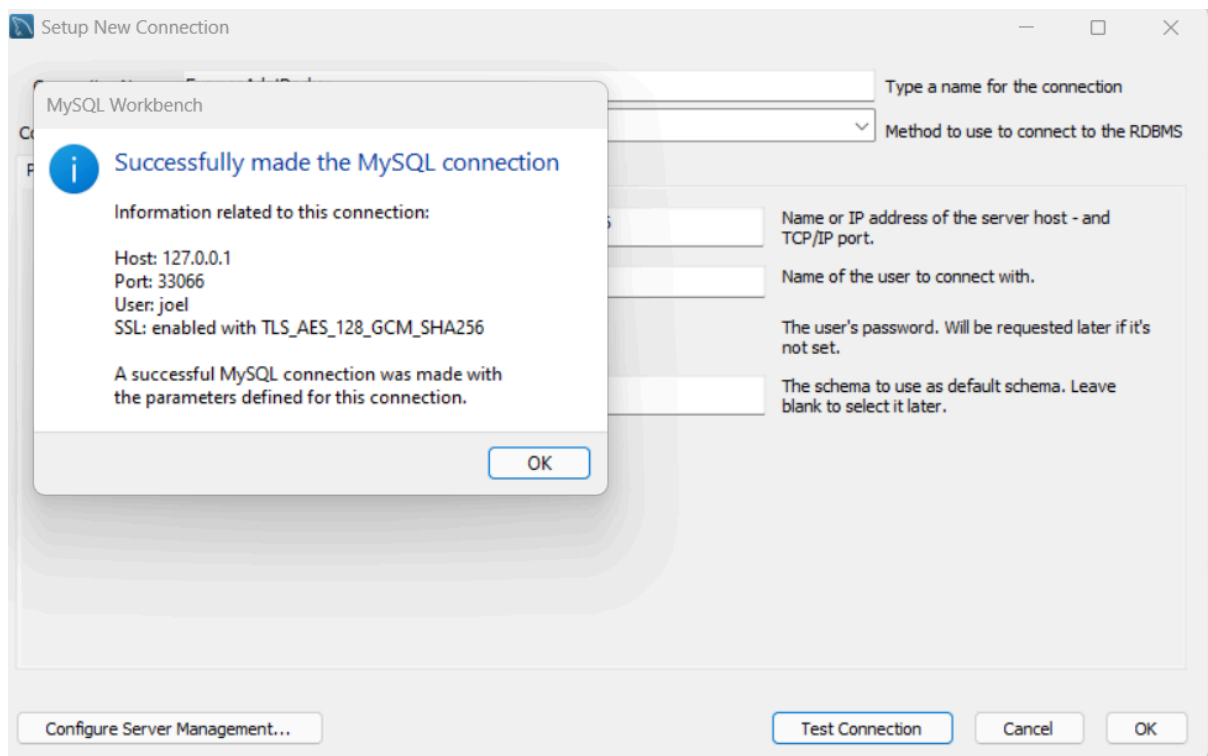
ahora vamos al workbench para conectarnos



## 9. CONECTARSE DESDE EL WORKBENCH



la contraseña es examen



## 10. CONECTARSE AL JDBC POR CODIGO

para ello, he creado un archivo JAVA llamado ConexionBBDD.java que es en el que contengo el código de esta actividad. Las líneas clave que modifique de un conector normal son las siguientes:

```
private static final String DB_URL =  
"jdbc:mysql://riomaiore-joel.cjqmcy2mujwe.us-east-1.rds.amazonaws.com:3  
306/";  
    private static final String USER = "admin";  
    private static final String PASSWORD = "12345678";
```

y el output muestra que se ha realizado correctamente:

```
PS C:\Users\HP\Desktop\DM2\ADAT\EXAMENES\T3\EXAMENJoe1\JavaProyectExamen>  
tb7moq5xmcdtjob.argfile' 'conexionBBDD.ConexionJDBC_AWS'  
Driver cargado correctamente.  
Conexión establecida con éxito!  
Conexión cerrada.  
PS C:\Users\HP\Desktop\DM2\ADAT\EXAMENES\T3\EXAMENJoe1\JavaProyectExamen>
```

## 11. CARGAR DATOS

The image shows a development environment with two main windows. The left window is an IDE (likely IntelliJ IDEA) showing a Java file named `Examen.java`. The code defines a `Examen` class with a `main` method that interacts with a database. The right window is MySQL Workbench, showing a query result for the `pedidosjoel` table.

**Java Code (Examen.java):**

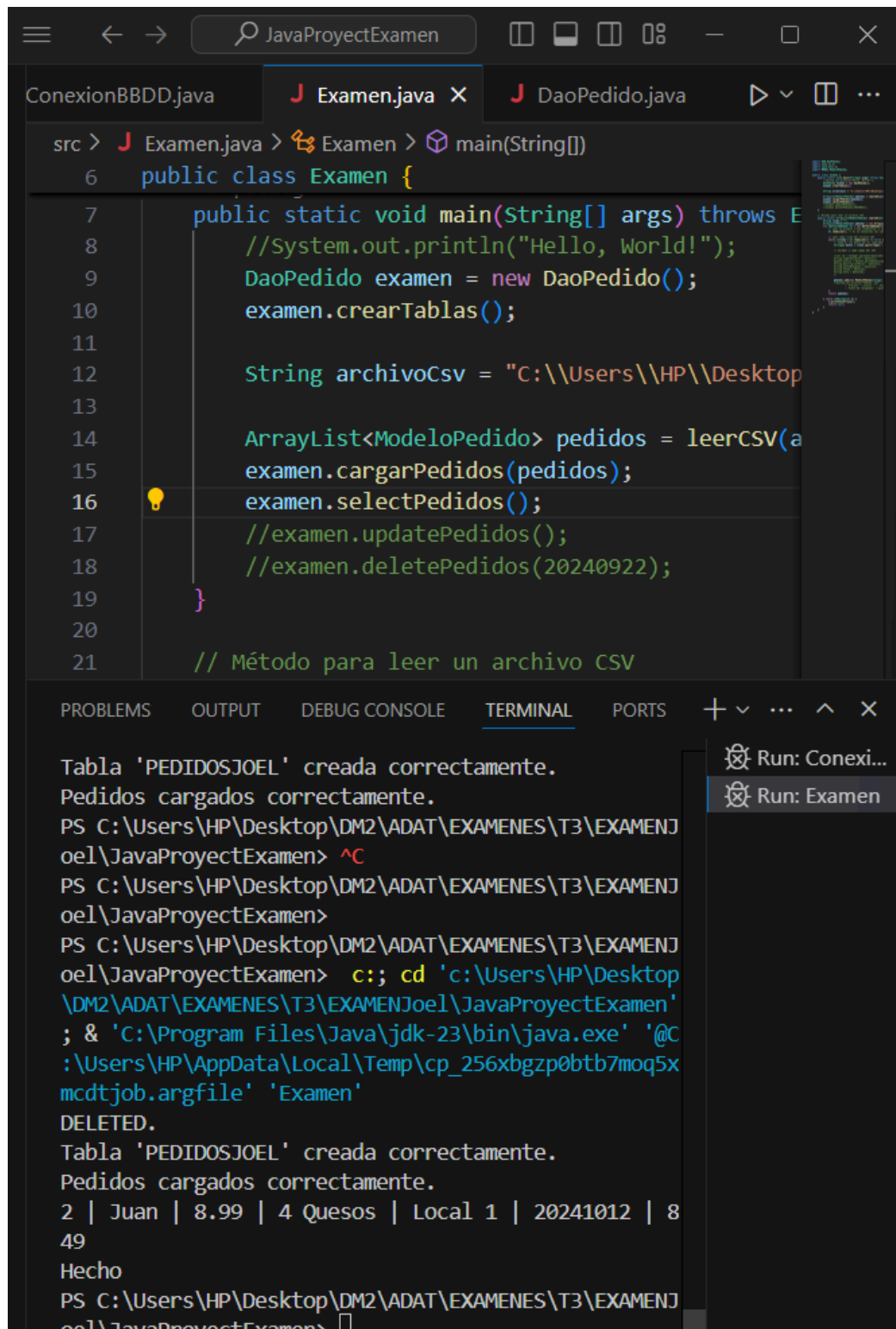
```
src > J Examen.java > Examen > main(String[])
4 import MODEL.MODELOPEDIDO;
5
6 public class Examen {
7     public static void main(String[] args) throws Exception {
8         //System.out.println("Hello, world!");
9         DaoPedido examen = new DaoPedido();
10        examen.crearTablas();
11
12        String archivoCsv = "c:\\Users\\VHP\\Desktop
13
14        ArrayList<ModeloPedido> pedidos = leerCSV(a
15        examen.cargarPedidos(pedidos);
16        //examen.selectPedidos();
17        //examen.updatePedidos();
18        //examen.deletePedidos(20240922);
19    }
20 }
```

**MySQL Workbench Query Result:**

Query 1: `SELECT * FROM pedidosjoel.pedidosjoel;`

ID	nombre_cliente	precio	descripcion_pizza	punto_recogida	fec
1	Carlos	9.75	Hawaiana	Local 1	202
2	Juan	8.99	4 Quesos	Local 3	202
3	Ara	10.25	Vegetal	Local 1	202
4	Pedro	9.75	Diavola	Local 1	202
5	Carlos	10.25	Carbonara	Local 2	202
6	Lucia	9.75	Carbonara	Local 1	202
7	Laura	12.5	Vegetal	Local 1	202
8	Pedro	13.5	Margarita	Local 1	202
9	Juan	9.75	Hawaiana	Local 1	202
10	Sofia	8.99	Barbacoa	Local 2	202
11	Luis	13.5	Carbonara	Local 3	202
12	Carlos	9.75	Margarita	Local 3	202
13	Miguel	11.2	Carbonara	Local 3	202
14	Lucia	8.99	Carbonara	Local 2	202

## 12. SELECT



```
src > J Examen.java > Examen > main(String[])
6 public class Examen {
7     public static void main(String[] args) throws E
8         //System.out.println("Hello, World!");
9         DaoPedido examen = new DaoPedido();
10        examen.crearTablas();
11
12        String archivoCsv = "C:\\Users\\HP\\Desktop
13
14        ArrayList<ModeloPedido> pedidos = leerCSV(a
15        examen.cargarPedidos(pedidos);
16        examen.selectPedidos();
17        //examen.updatePedidos();
18        //examen.deletePedidos(20240922);
19    }
20
21    // Método para leer un archivo CSV
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + v ... ^ x

Tabla 'PEDIDOSJOEL' creada correctamente.  
Pedidos cargados correctamente.  
PS C:\Users\HP\Desktop\DM2\ADAT\EXAMENES\T3\EXAMENJ  
oel\JavaProyectExamen> ^C  
PS C:\Users\HP\Desktop\DM2\ADAT\EXAMENES\T3\EXAMENJ  
oel\JavaProyectExamen>  
PS C:\Users\HP\Desktop\DM2\ADAT\EXAMENES\T3\EXAMENJ  
oel\JavaProyectExamen> c:; cd 'c:\Users\HP\Desktop  
\DM2\ADAT\EXAMENES\T3\EXAMENJoel\JavaProyectExamen'  
; & 'C:\Program Files\Java\jdk-23\bin\java.exe' '@C  
:\Users\HP\AppData\Local\Temp\cp\_256xbgzp0btb7moq5x  
mcdtjob.argfile' 'Examen'  
DELETED.  
Tabla 'PEDIDOSJOEL' creada correctamente.  
Pedidos cargados correctamente.  
2 | Juan | 8.99 | 4 Quesos | Local 1 | 20241012 | 8  
49  
Hecho  
PS C:\Users\HP\Desktop\DM2\ADAT\EXAMENES\T3\EXAMENJ  
oel\JavaProyectExamen>

Run: Conexi...  
Run: Examen

## 13. UPDATE

The screenshot shows the MySQL Workbench interface. On the left, the 'Navigation' pane shows the 'pedidosjoel' database and its tables. The 'Table: pedidosjoel' is selected, showing its structure with columns: ID (decimal), nombre\_cliente (varchar), precio (decimal), descripcion\_pizza (varchar), and punto\_recogida (varchar). The 'Columns' tab is active, showing the table's schema. On the right, the 'SQL' editor shows a query: `SELECT * FROM pedidosjoel.pedidosjoel;`. The 'Results' pane shows the query results, displaying 14 rows of data. The 'Table: pedidosjoel' is selected, showing its structure with columns: ID (decimal), nombre\_cliente (varchar), precio (decimal), descripcion\_pizza (varchar), and punto\_recogida (varchar). The 'Columns' tab is active, showing the table's schema. On the right, the 'SQL' editor shows a query: `SELECT * FROM pedidosjoel.pedidosjoel;`. The 'Results' pane shows the query results, displaying 14 rows of data.

## 14. DELETE

The screenshot displays a development environment with two main windows. The left window is an IDE showing a Java file named `Examen.java`. The code defines a `main` method that interacts with a database, including a `deletePedidos` call. The right window is MySQL Workbench, showing a query editor with a `SELECT` statement and a result grid displaying data from the `pedidosjoel` table.

**Java Code (Examen.java):**

```
src > J Examen.java > Examen > main(String[]) throws E
6 public class Examen {
7     public static void main(String[] args) throws E
16         examen.selectPedidos();
17         examen.updatePedidos();
18         examen.deletePedidos(dia:20240922);
19     }
20 // Método para leer un archivo CSV
21 }
```

**MySQL Workbench Query:**

```
SELECT * FROM pedidosjoel.pedidosjoel;
```

**MySQL Workbench Result Grid:**

ID	nombre_cliente	precio	descripcion_pizza	punto
1	Carlos	11.7	Hawaiana	Local 1
2	Juan	10.788	4 Quesos	Local 1
3	Ana	12.299999999999999	Vegetal	Local 3
5	Carlos	12.299999999999999	Carbonara	Local 2
6	Luca	11.7	Carbonara	Local 1
7	Laura	15	Vegetal	Local 1
8	Pedro	16.2	Margarita	Local 1
9	Juan	11.7	Hawaiana	Local 1
10	Sofia	10.788	Barbacoa	Local 2
11	Luis	16.2	Carbonara	Local 3
13	Miguel	13.44	Carbonara	Local 3
14	Lucia	10.788	Carbonara	Local 2
15	Juan	12.299999999999999	Diavola	Local 3
16	Laura	16.2	Barbacoa	Local 2

**MySQL Workbench Information Panel:**

Table: **pedidosjoel**

Columns:

- ID: decimal PK
- nombre\_cliente: varchar
- precio: double
- descripcion\_pizza: varchar
- punto\_recogida: varchar