

EJERCICIO TRANSACCIONES



JDBC
SQL con Java



Richard Chadwick Plaza

Acceso a Datos

11/11/2024 2ºDAM

1. CÓDIGO SQL. ESTRUCTURA DE LA BASE DE DATOS

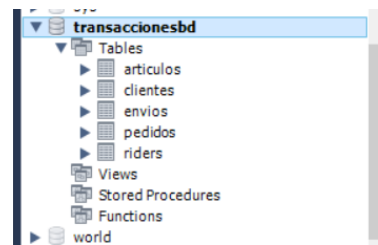
```
ejercicioTransacciones  transaccionesBD x mi_base_datos  clientes  articulos  pedidos
-- Creamos la base de datos
1 • CREATE database transaccionesBD;
2
3
4 -- Nos aseguramos que utilizamos esta base de datos
5 • USE transaccionesBD;
6
7 -- Tabla articulos
8 • CREATE TABLE articulos(
9     codart INT AUTO_INCREMENT PRIMARY KEY,
10     descripcion VARCHAR(100),
11     existencias INT,
12     precio DECIMAL(10, 2)
13 );
14
15 -- Tabla clientes, esta tabla no la pedía el enunciado, pero creo que el programa que
16 • CREATE TABLE clientes(
17     codcli INT AUTO_INCREMENT PRIMARY KEY,
18     nombre VARCHAR(100),
19     direccion VARCHAR(100),
20     telefono VARCHAR(15)
21 );
22
23
```

```
ejercicioTransacciones  transaccionesBD x mi_base_datos  clientes  articulos  pedidos
25 • CREATE TABLE pedidos(
26     codped INT AUTO_INCREMENT PRIMARY KEY,
27     fecha DATE,
28     codcli INT,
29     direccion VARCHAR(100),
30     codart INT,
31     FOREIGN KEY(codart) REFERENCES articulos(codart),
32     FOREIGN KEY(codcli) REFERENCES clientes(codcli)
33 );
34
35 -- Tabla de riders
```

```
ejercicioTransacciones  transaccionesBD x  mi_base_datos  clientes  articulos  pedidos

34
35  -- Tabla de riders
36  CREATE TABLE riders(
37      codrider INT AUTO_INCREMENT PRIMARY KEY,
38      nombre VARCHAR(100),
39      disponibilidad INTEGER CHECK (disponibilidad IN (1, 0))  -- 1 representa que e
40  );
41
42  -- Tabla de Envios
43  CREATE TABLE envios(
44      codped INT,
45      codrider INT,
46      terminado DATETIME,
47      PRIMARY KEY (codped, codrider),
48      FOREIGN KEY (codped) REFERENCES pedidos(codped),
49      FOREIGN KEY (codrider) REFERENCES riders(codrider)
50  );
51
```

Estructura de la base de datos:

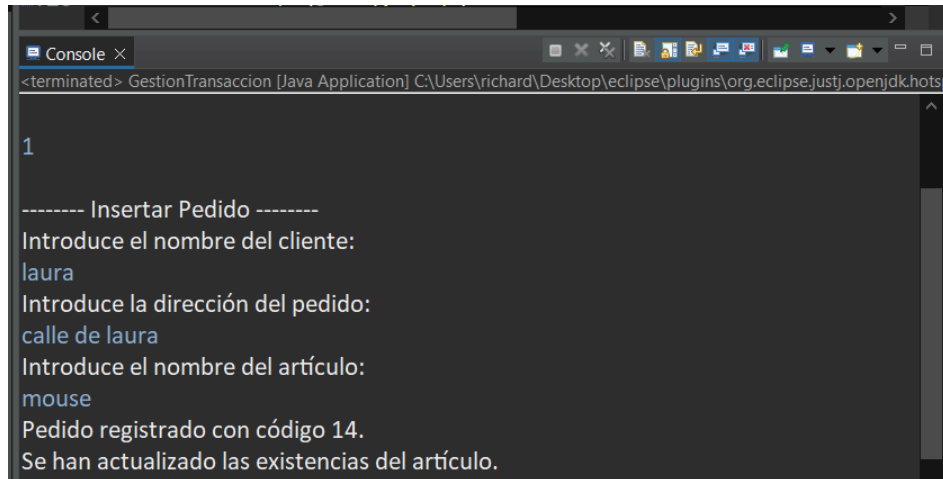


Voy a insertar algunos campos en las tablas:

```
--
52  -- Datos para la tabla articulos
53  INSERT INTO articulos (descripcion, existencias, precio)
54  VALUES
55      ('Laptop', 50, 800.00),
56      ('Mouse', 200, 20.00),
57      ('Teclado', 150, 50.00);
58
59  -- Datos para la tabla riders
60  INSERT INTO riders (nombre, disponibilidad)
61  VALUES
62      ('Richard Chadwick', 1),
63      ('Hugo Jiménez', 1),
64      ('Eleazar Ballesteros', 1);
65
66  -- Datos iniciales de clientes
67  INSERT INTO clientes(nombre, direccion, telefono) VALUES
68      ('Alberto', 'calle de alberto', '123456789'),
69      ('Laura', 'calle de Laura', '098765432'),
70      ('Jose', 'calle de Jose', '123443212');
71
72
```

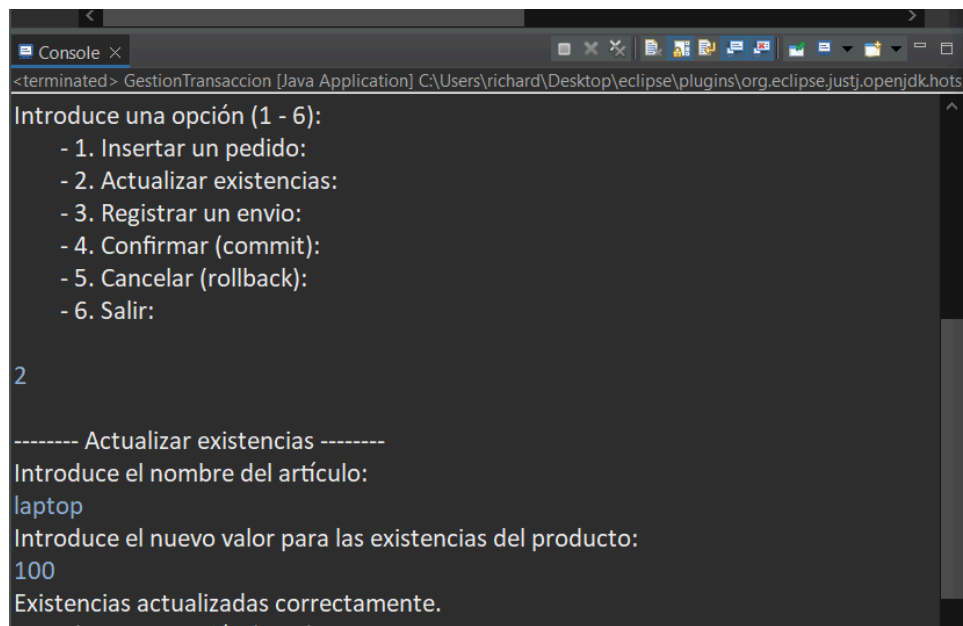
2. Ejecución del código

Probamos a insertar un pedido



```
<terminated> GestionTransaccion [Java Application] C:\Users\richard\Desktop\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full\jre\bin\java.exe
1
----- Insertar Pedido -----
Introduce el nombre del cliente:
laura
Introduce la dirección del pedido:
calle de laura
Introduce el nombre del artículo:
mouse
Pedido registrado con código 14.
Se han actualizado las existencias del artículo.
```

Actualizamos las existencias del producto laptop:



```
<terminated> GestionTransaccion [Java Application] C:\Users\richard\Desktop\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full\jre\bin\java.exe
Introduce una opción (1 - 6):
- 1. Insertar un pedido:
- 2. Actualizar existencias:
- 3. Registrar un envío:
- 4. Confirmar (commit):
- 5. Cancelar (rollback):
- 6. Salir:
2
----- Actualizar existencias -----
Introduce el nombre del artículo:
laptop
Introduce el nuevo valor para las existencias del producto:
100
Existencias actualizadas correctamente.
```

Registramos el envío:

```
Introduce una opción (1 - 6):
- 1. Insertar un pedido:
- 2. Actualizar existencias:
- 3. Registrar un envío:
- 4. Confirmar (commit):
- 5. Cancelar (rollback):
- 6. Salir:

3

----- Registrar envío -----
Envío registrado correctamente.
Se ha actualizado el rider a no disponible
```

Probamos a hacer el commit:

```
Introduce una opción (1 - 6):
- 1. Insertar un pedido:
- 2. Actualizar existencias:
- 3. Registrar un envío:
- 4. Confirmar (commit):
- 5. Cancelar (rollback):
- 6. Salir:

4

Transacción confirmada exitosamente.
El pedido está en reparto
El rider 3 está entregando el pedido 14
Se ha marcado el pedido 14 como entregado
El rider 3 vuelve a estar disponible al haber terminado la entrega del pedido
El envío del pedido 14 ha sido marcado como terminado.
```

En la tabla de artículos comprobamos que se han actualizado las existencias del laptop:

The screenshot shows the SQL Server Enterprise interface with the 'transaccionesbd' database selected. The 'articulos' table is open, displaying the following data:

codart	descripcion	existencias	precio
1	Laptop	100	800.00
2	Mouse	99	20.00
3	Teclado	150	50.00

The 'Output' pane shows the execution of three queries:

- 20 16:36:26 SELECT * FROM transaccionesbd.envios; 7 row(s) returned
- 21 16:36:27 SELECT * FROM transaccionesbd.pedidos; 9 row(s) returned
- 22 16:36:28 SELECT * FROM transaccionesbd.riders; 3 row(s) returned

En la tabla de pedidos podemos ver el último pedido que he hecho, es el del codped = 14

The screenshot shows the SQL Server Enterprise interface with the 'transaccionesbd' database selected. The 'pedidos' table is open, displaying the following data:

codped	fecha	codcli	direccion	codart
11	2024-11-18	2	calle de laura	1
12	2024-11-18	1	calle de alberto	1
13	2024-11-18	2	calle de laura	1
14	2024-11-18	2	calle de laura	2

The 'Output' pane shows the execution of three queries:

- 20 16:36:26 SELECT * FROM transaccionesbd.envios; 7 row(s) returned
- 21 16:36:27 SELECT * FROM transaccionesbd.pedidos; 9 row(s) returned
- 22 16:36:28 SELECT * FROM transaccionesbd.riders; 3 row(s) returned

En la tabla envío se ve el pedido y el rider que se le ha asociado, aparece la fecha en la que se ha terminado el reparto:

The screenshot shows the SQL Server Enterprise Manager interface with the 'envios' table selected. The 'Result Grid' displays the following data:

codped	codrider	terminado
11	1	NULL
12	2	NULL
13	2	NULL
14	3	2024-11-18 16:35:43

The 'Output' window shows the execution of three SQL queries:

#	Time	Action	Message	Duration / Fetch
20	16:36:26	SELECT * FROM transaccionesbd.envios	7 row(s) returned	0.000 sec / 0.000 sec
21	16:36:27	SELECT * FROM transaccionesbd.pedidos	9 row(s) returned	0.000 sec / 0.000 sec
22	16:36:28	SELECT * FROM transaccionesbd.riders	3 row(s) returned	0.000 sec / 0.000 sec

En la tabla rider aparece que el rider 3, eleazar está disponible porque ha terminado el reparto:

The screenshot shows the SQL Server Enterprise Manager interface with the 'riders' table selected. The 'Result Grid' displays the following data:

codrider	nombre	disponibilidad
1	Richard Chadwick	0
2	Hugo Jiménez	0
3	Eleazar Ballesteros	1

The 'Output' window shows the execution of three SQL queries:

#	Time	Action	Message	Duration / Fetch
20	16:36:26	SELECT * FROM transaccionesbd.envios	7 row(s) returned	0.000 sec / 0.000 sec
21	16:36:27	SELECT * FROM transaccionesbd.pedidos	9 row(s) returned	0.000 sec / 0.000 sec
22	16:36:28	SELECT * FROM transaccionesbd.riders	3 row(s) returned	0.000 sec / 0.000 sec