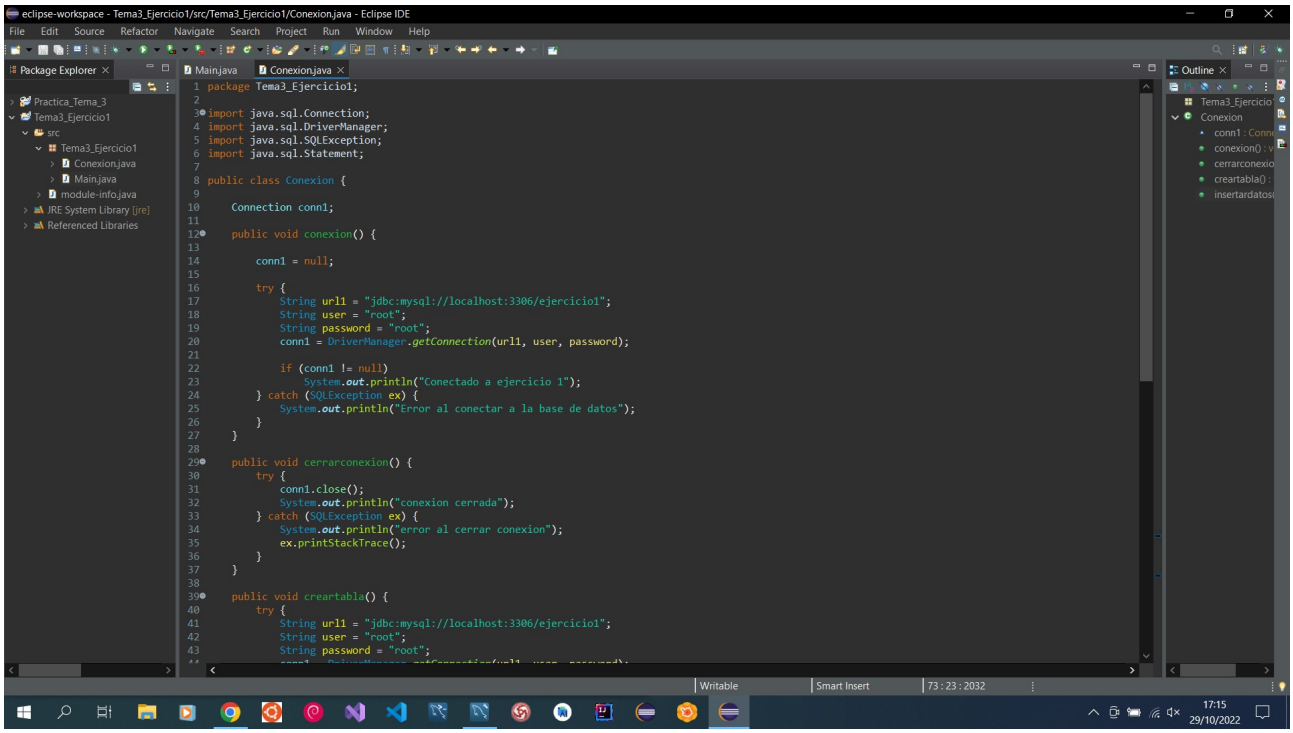
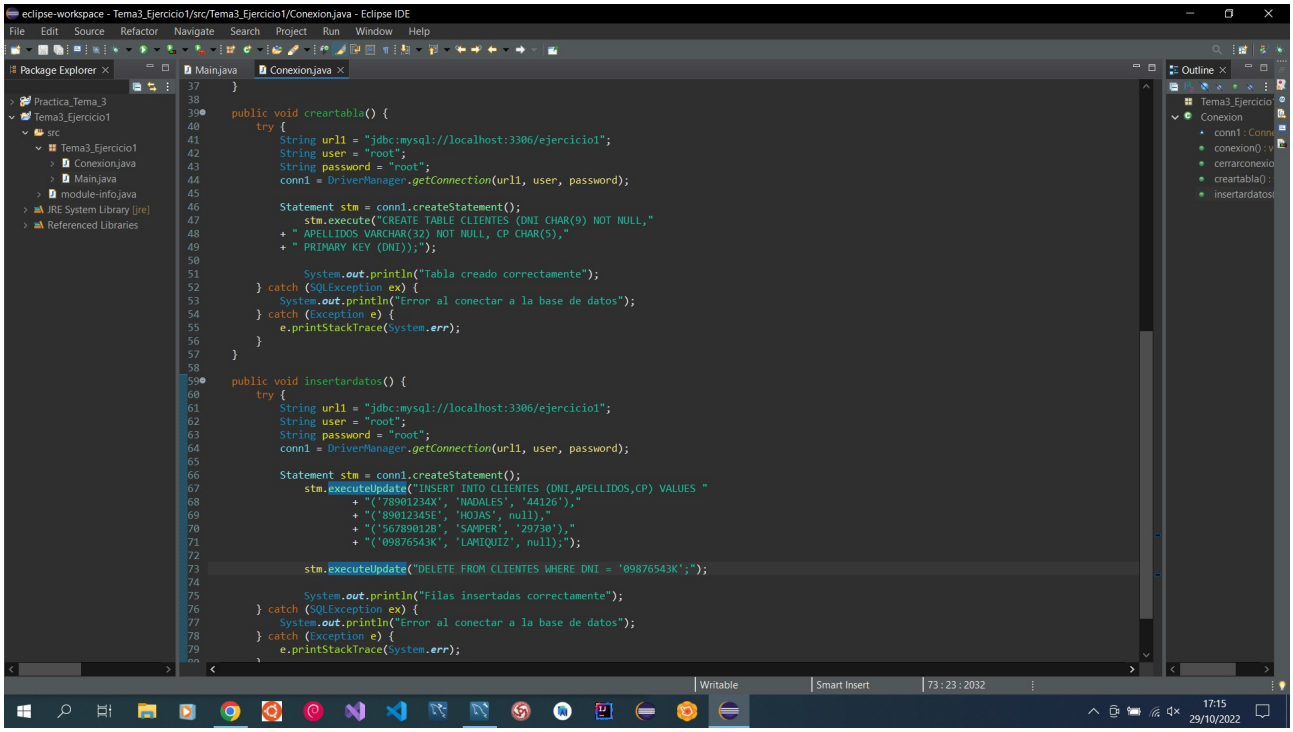
Ejercicios Tema 3: BBDD Relacionales

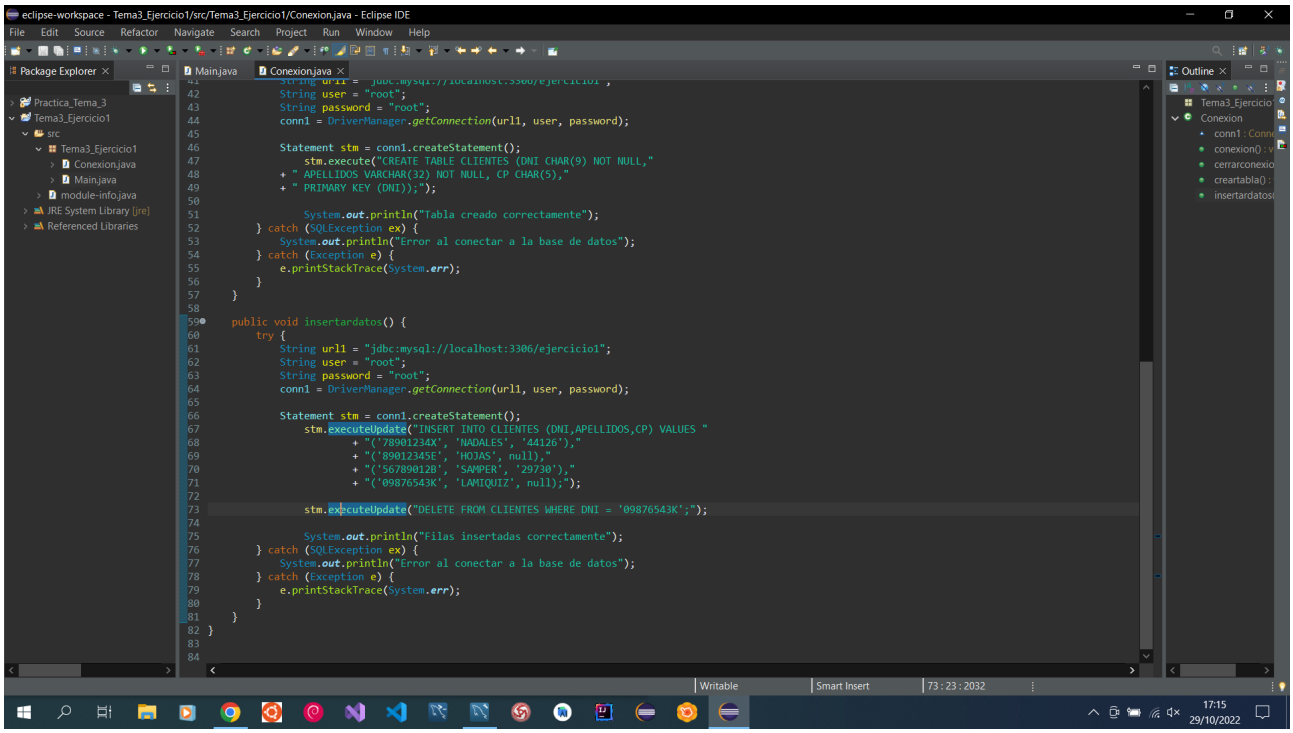
Adjunto imágenes del programa y su ejecución, donde también mandaré el proyecto junto con el PDF.

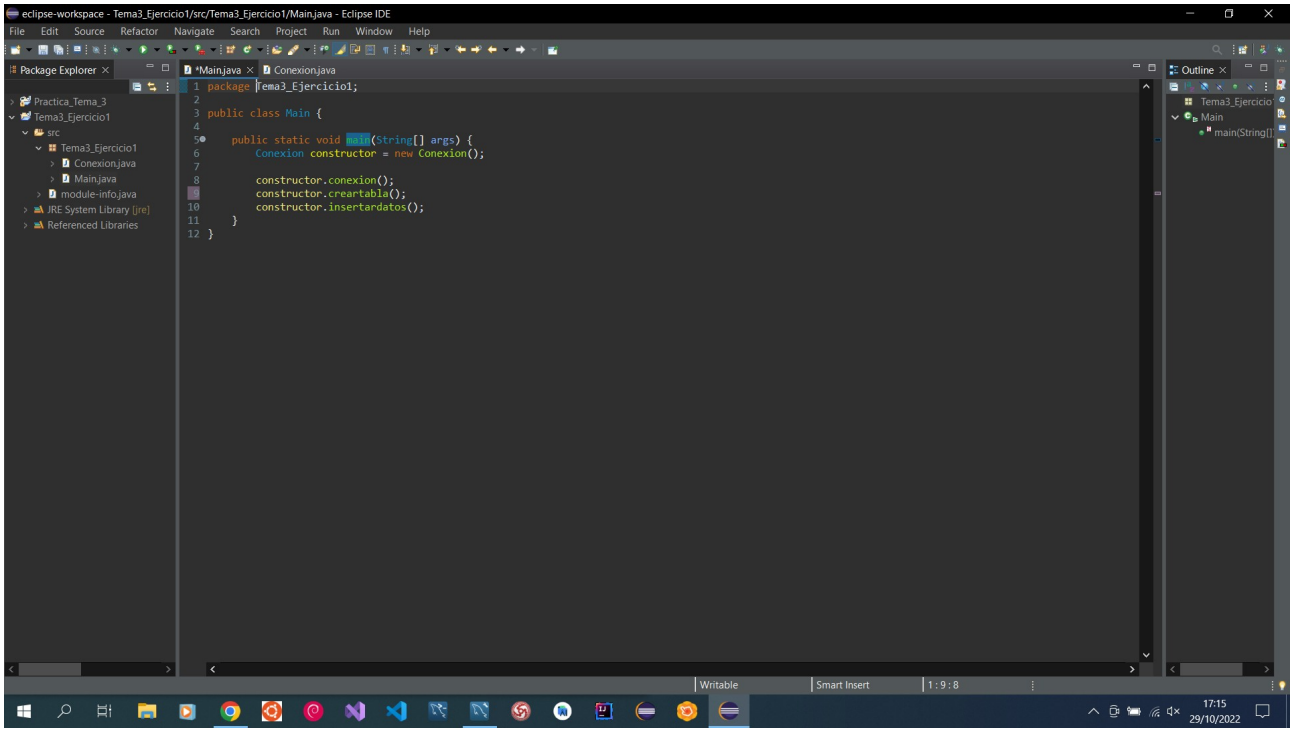
3.4 – JAVA DATABASE CONNECTIVITY (JDBC)

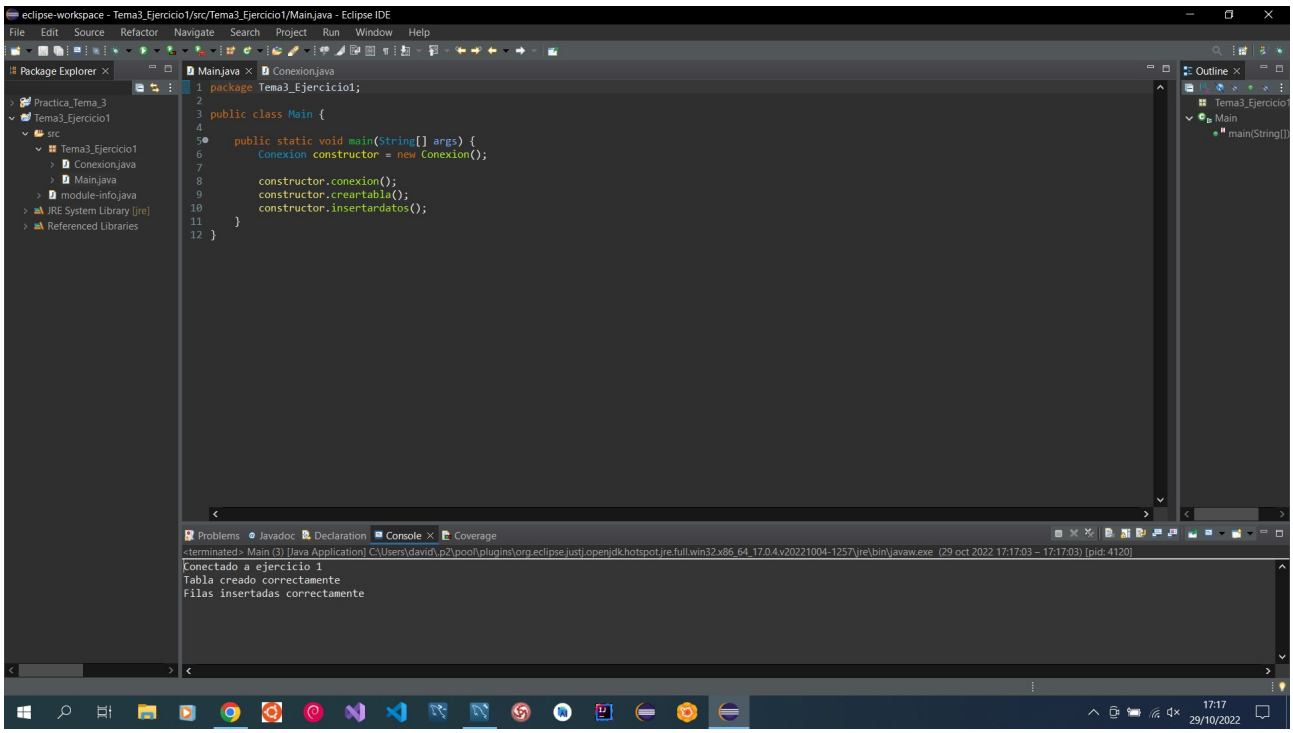
a) Haz un programa que haga los cambios necesarios para que los contenidos de la tabla CLIENTES sean los siguientes: (‘78901234X’, ‘NADALES’, ‘44126’), (‘89012345E’, ‘ROJAS’, null), (‘56789012B’, ‘SAMPER’, ‘29730’), partiendo de los contenidos de la tabla resultantes de la ejecución del ejemplo anterior. El programa debe utilizar sentencias UPDATE y DELETE.

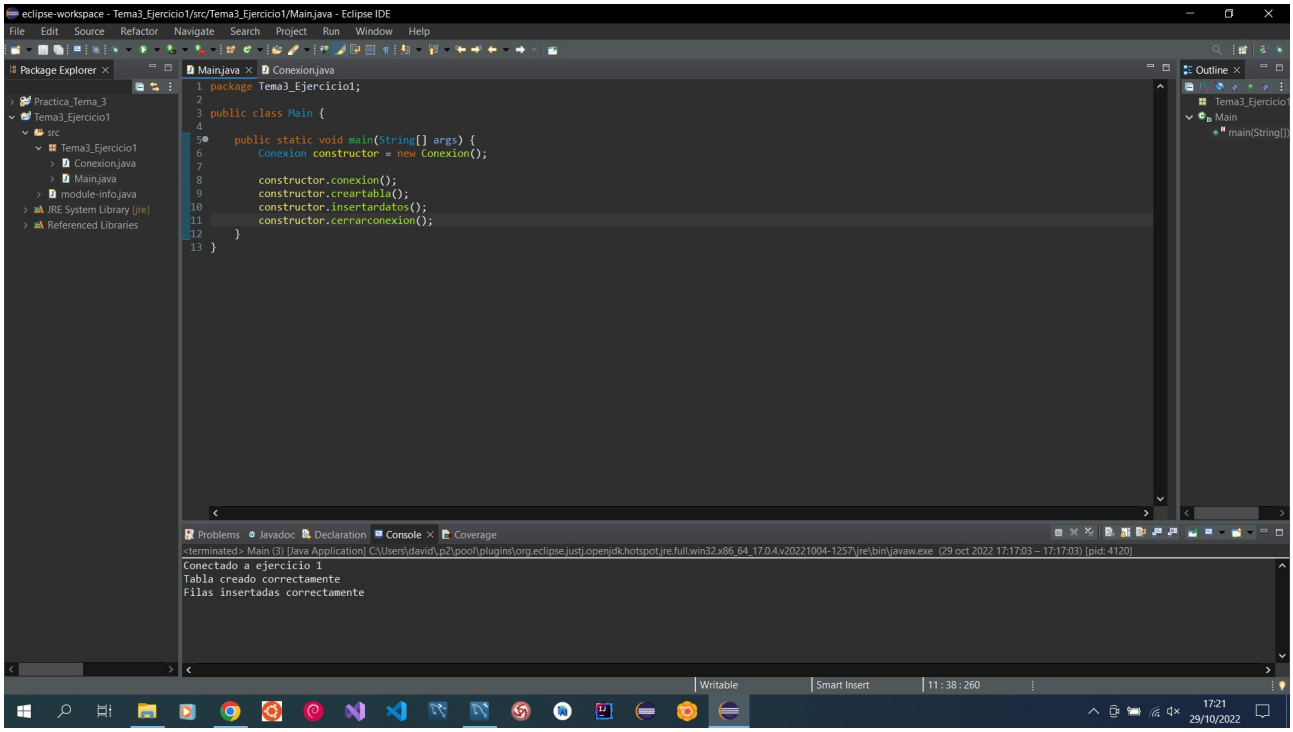


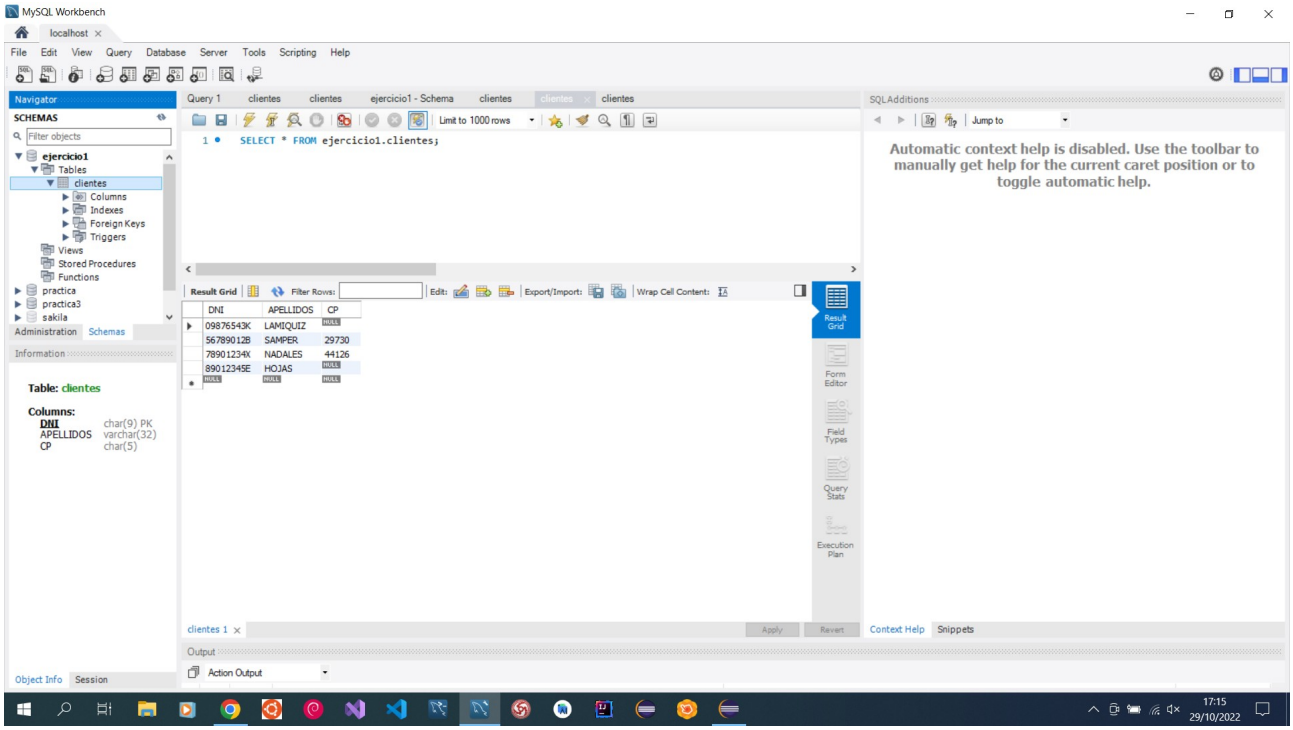

eclipse-workspace - Tema3_Ejercicio1/src/Tema3_Ejercicio1/Conexion.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Main.java
Conexion.java
Outline
Practica_Tema_3
Tema3_Ejercicio1
src
Tema3_Ejercicio1
Conexion.java
Main.java
module-info.java
JRE System Library [jre]
Referenced Libraries
Writable
Smart Insert
73 : 23 : 2032
17:15
29/10/2022

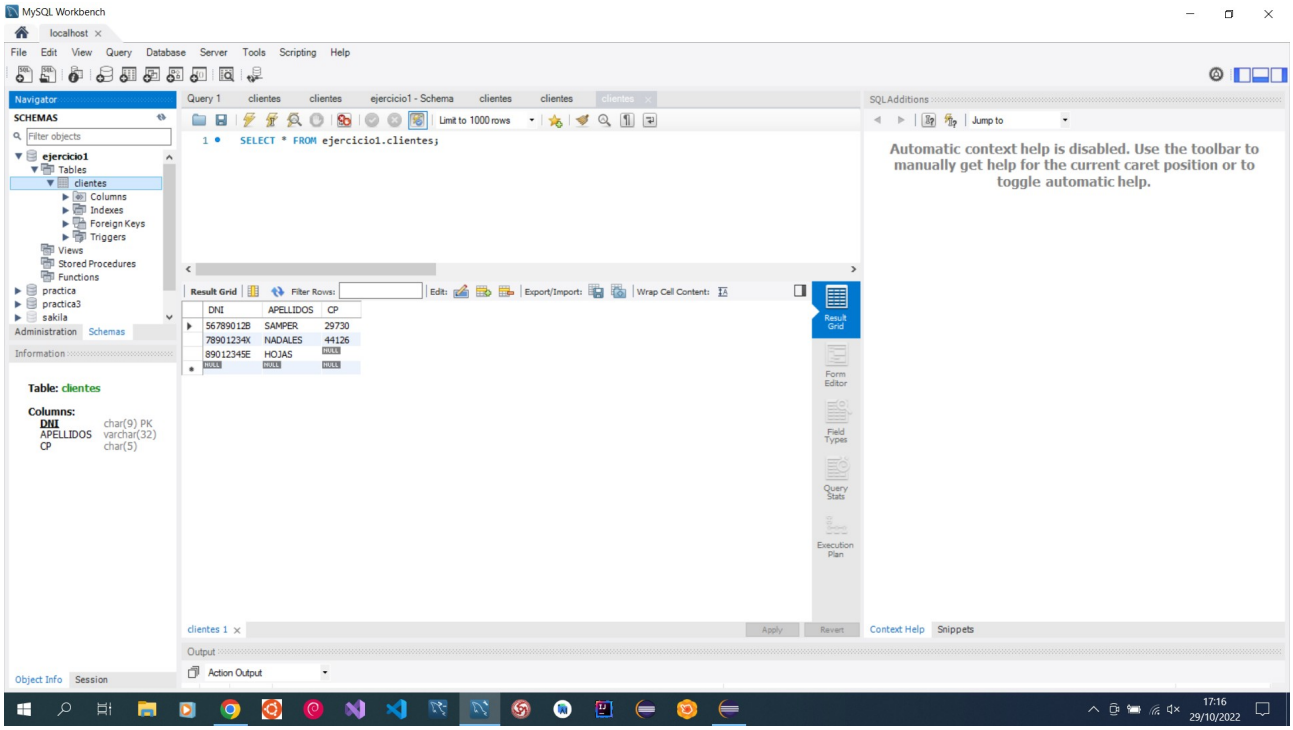

eclipse-workspace - Tema3_Ejercicio1/src/Tema3_Ejercicio1/Conexion.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Main.java
Conexion.java
Outline
Practica_Tema_3
Tema3_Ejercicio1
src
Tema3_Ejercicio1
Conexion.java
Main.java
module-info.java
JRE System Library [jre]
Referenced Libraries
Writable
Smart Insert
73 : 23 : 2032
17:15
29/10/2022


eclipse-workspace - Tema3_Ejercicio1/src/Tema3_Ejercicio1/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Practica_Tema_3
Tema3_Ejercicio1
src
Tema3_Ejercicio1
Conexion.java
Main.java
module-info.java
JRE System Library [jre]
Referenced Libraries
*Main.java
Conexion.java
1 package Tema3_Ejercicio1;
2
3 public class Main {
4
5 public static void main(String[] args) {
6 Conexion constructor = new Conexion();
7
8 constructor.conexion();
9 constructor.creartabla();
10 constructor.insertardatos();
11 }
12 }
Outline
Tema3_Ejercicio1
Main
main(String[])
Writable
Smart Insert
1 : 9 : 8
17:15
29/10/2022


eclipse-workspace - Tema3_Ejercicio1/src/Tema3_Ejercicio1/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Practica_Tema_3
Tema3_Ejercicio1
src
Tema3_Ejercicio1
Conexion.java
Main.java
module-info.java
JRE System Library [jre]
Referenced Libraries
Main.java
Conexion.java
1 package Tema3_Ejercicio1;
2
3 public class Main {
4
5 public static void main(String[] args) {
6 Conexion constructor = new Conexion();
7
8 constructor.conexion();
9 constructor.creartabla();
10 constructor.insertardatos();
11 }
12 }
Outline
Tema3_Ejercicio1
Main
main(String[])
Problems Javadoc Declaration Console Coverage
<terminated> Main (3) [Java Application] C:\Users\david\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20221004-1257\jre\bin\javaw.exe (29 oct 2022 17:17:03 – 17:17:03) [pid: 4120]
Conectado a ejercicio 1
Tabla creado correctamente
Filas insertadas correctamente
17:17
29/10/2022

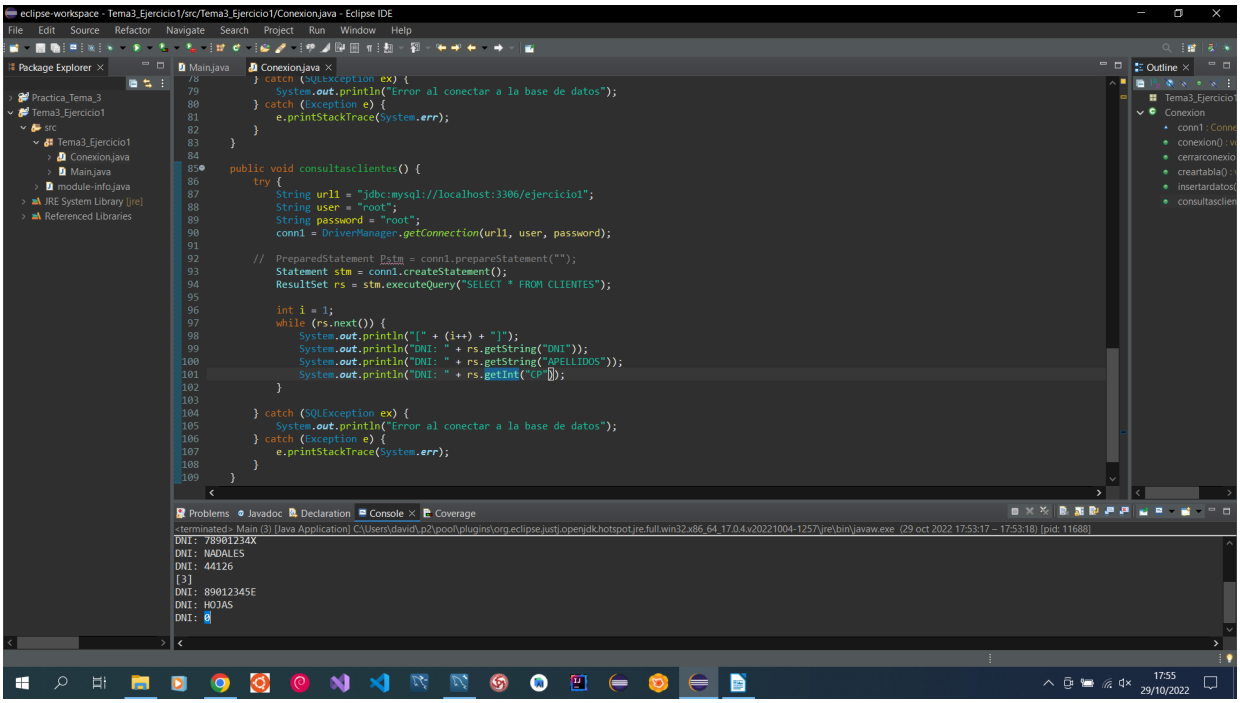

eclipse-workspace - Tema3_Ejercicio1/src/Tema3_Ejercicio1/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Practica_Tema_3
Tema3_Ejercicio1
src
Tema3_Ejercicio1
Conexion.java
Main.java
module-info.java
JRE System Library [jre]
Referenced Libraries
Main.java
Conexion.java
1 package Tema3_Ejercicio1;
2
3 public class Main {
4
5 public static void main(String[] args) {
6 Conexion constructor = new Conexion();
7
8 constructor.conexion();
9 constructor.creartabla();
10 constructor.insertardatos();
11 constructor.cerrarconexion();
12 }
13 }
Outline
Tema3_Ejercicio1
Main
main(String[])
Problems Javadoc Declaration Console Coverage
<terminated> Main (3) [Java Application] C:\Users\david\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20221004-1257\jre\bin\javaw.exe (29 oct 2022 17:17:03 – 17:17:03) [pid: 4120]
Conectado a ejercicio 1
Tabla creado correctamente
Filas insertadas correctamente
Writable
Smart Insert
11 : 38 : 260
17:21
29/10/2022


MySQL Workbench
localhost
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
ejercicio1
Tables
clientes
Columns
Indexes
Foreign Keys
Triggers
Views
Stored Procedures
Functions
practica
practica3
sakila
Administration Schemas
Information
Table: clientes
Columns:
DNI char(9) PK
APELLIDOS varchar(32)
CP char(5)
Query 1 clientes clientes ejercicio1 - Schema clientes clientes clientes
Limit to 1000 rows
1 SELECT * FROM ejercicio1.clientes;
Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Contents:
DNI APELLIDOS CP
09876543K LAMIQUIZ NULL
56789012B SAMPER 29730
78901234X NADALES 44126
89012345E HOJAS NULL
NULL NULL NULL
clientes 1
Apply Revert
Output
Action Output
Object Info Session
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Context Help Snippets
17:15
29/10/2022



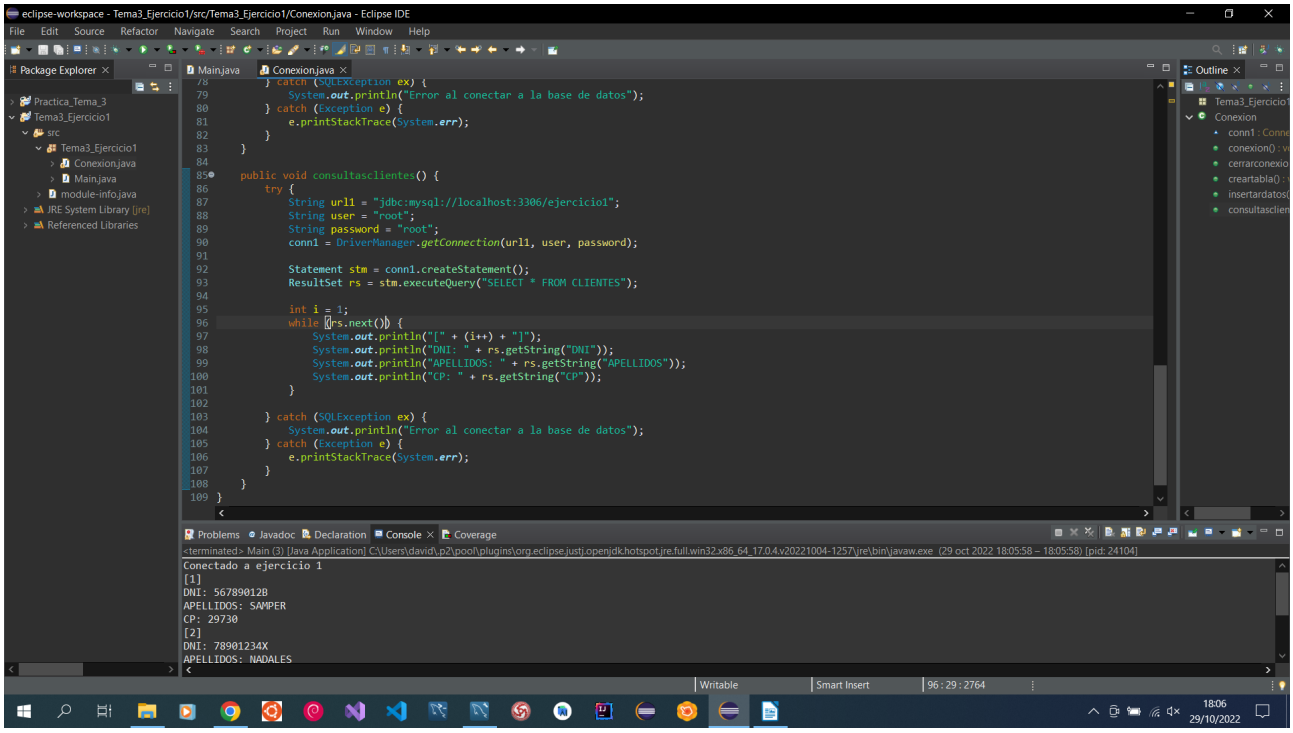
b) El código postal (columna CP) está definido con tipo CHAR(5) en la tabla CLIENTES, pero es siempre un número entero. ¿Se podría utilizar getInt() en lugar de getString() para recuperar su valor? Cambia el programa y verifica tu hipótesis, o justifica los resultados si no son los que esperabas.

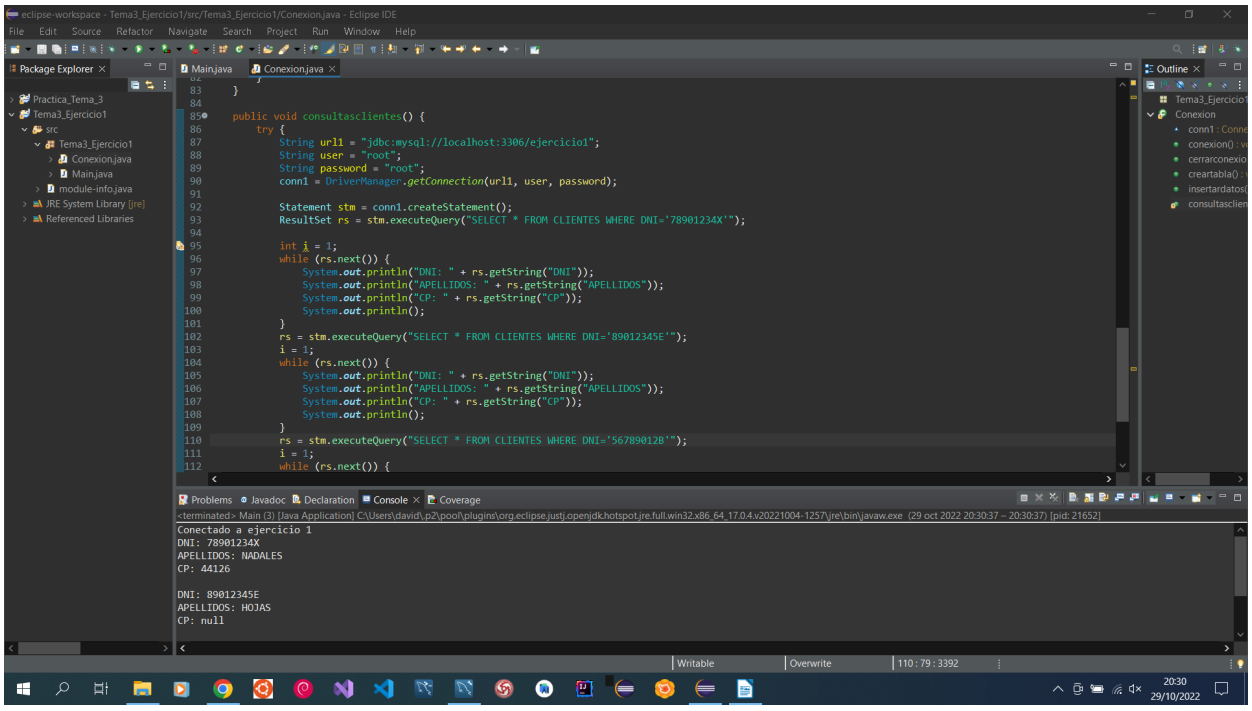
Se puede realizar y funciona pero no debidamente, dado que su valor real es un Char; un carácter puede tener valor null mientras que un Integer no hay valor null y cuando el char es null saca el valor Integer “0”.

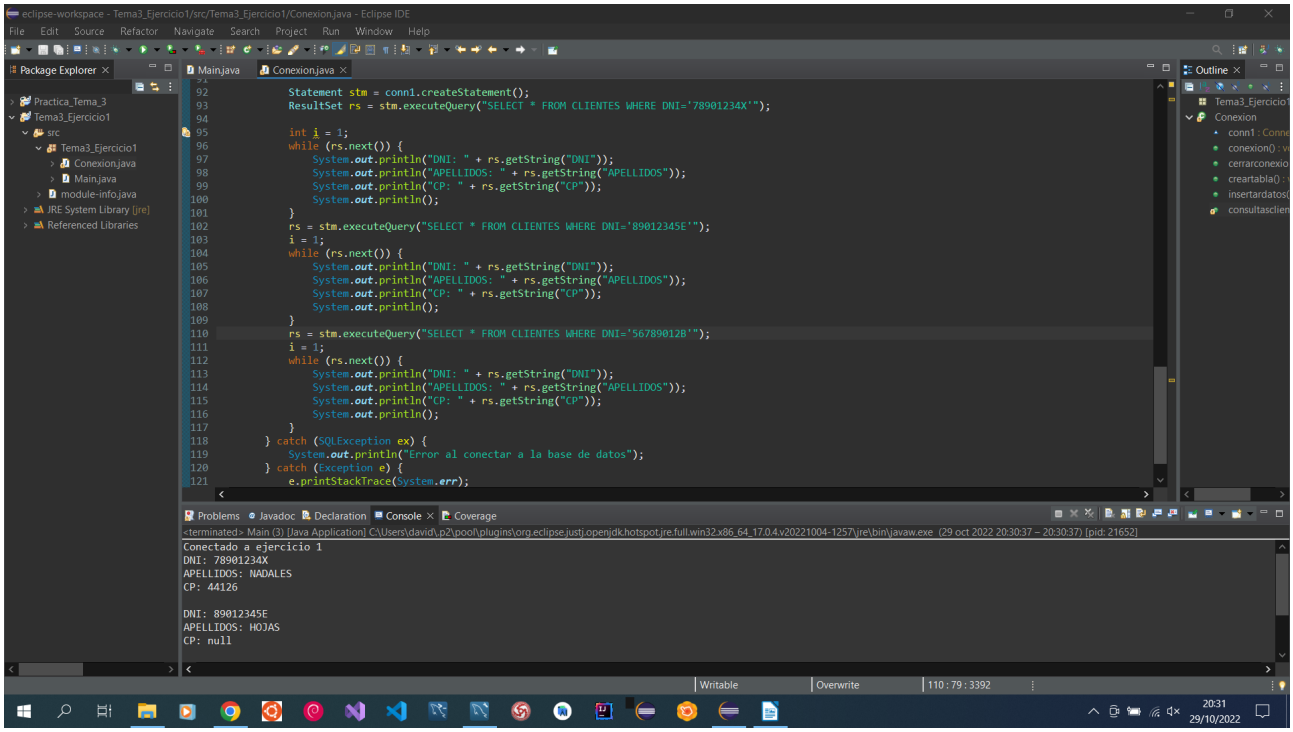


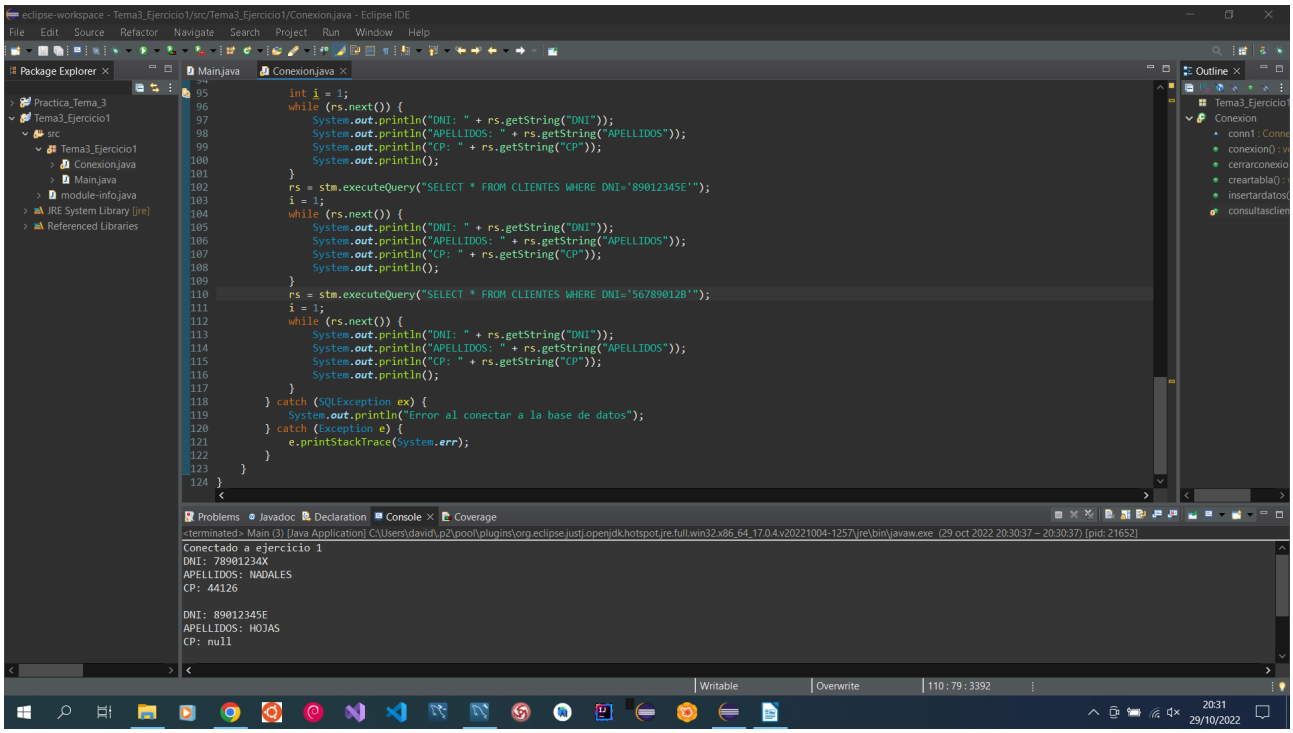
3.5 – SENTENCIAS PREPARADAS, TRANSACCIONES Y CLAVES AUTOGENERADAS

a) Escribe un programa que muestre los datos de varios clientes, o todos, de la tabla CLIENTES1. El programa debe utilizar una sentencia preparada para la consulta SELECT * FROM CLIENTES1 WHERE DNI=? Debe realizarse una consulta para cada cliente, especificando su DNI, y obtener los datos del ResultSet resultante, que solo tendrá una fila, al ser el acceso por clave primaria.

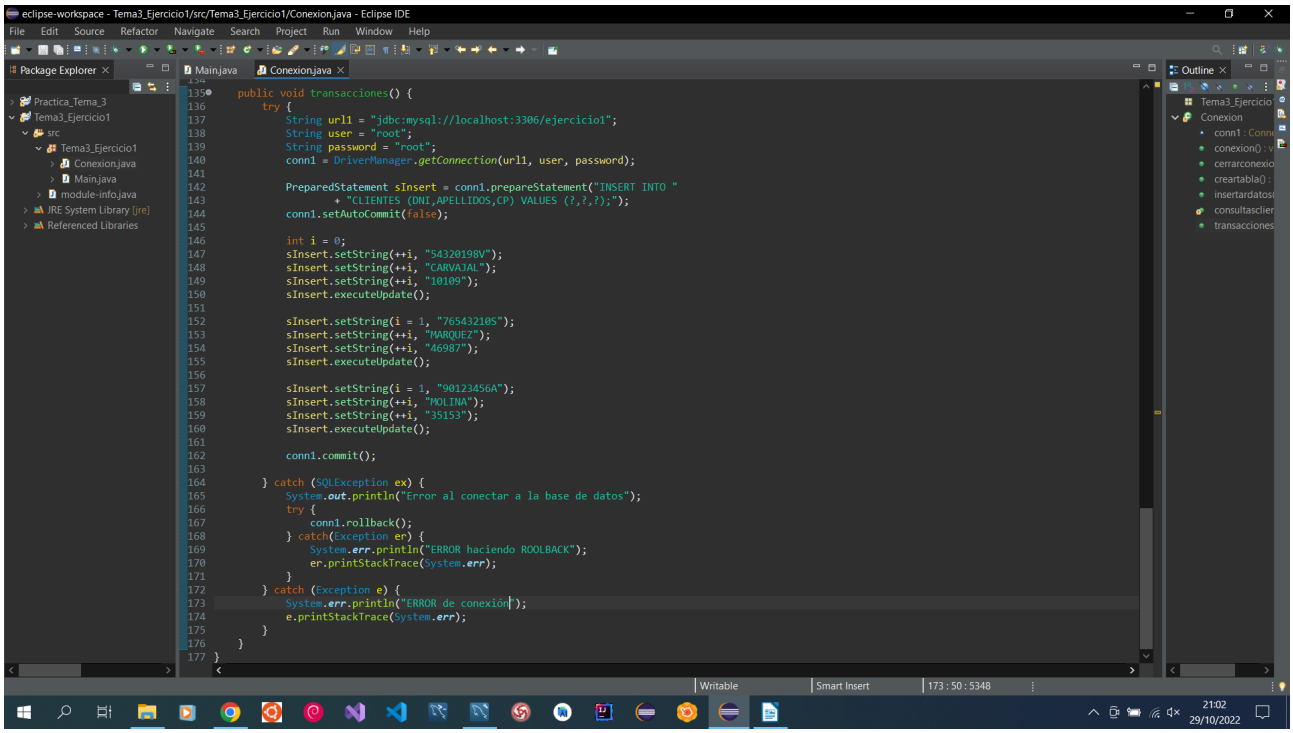


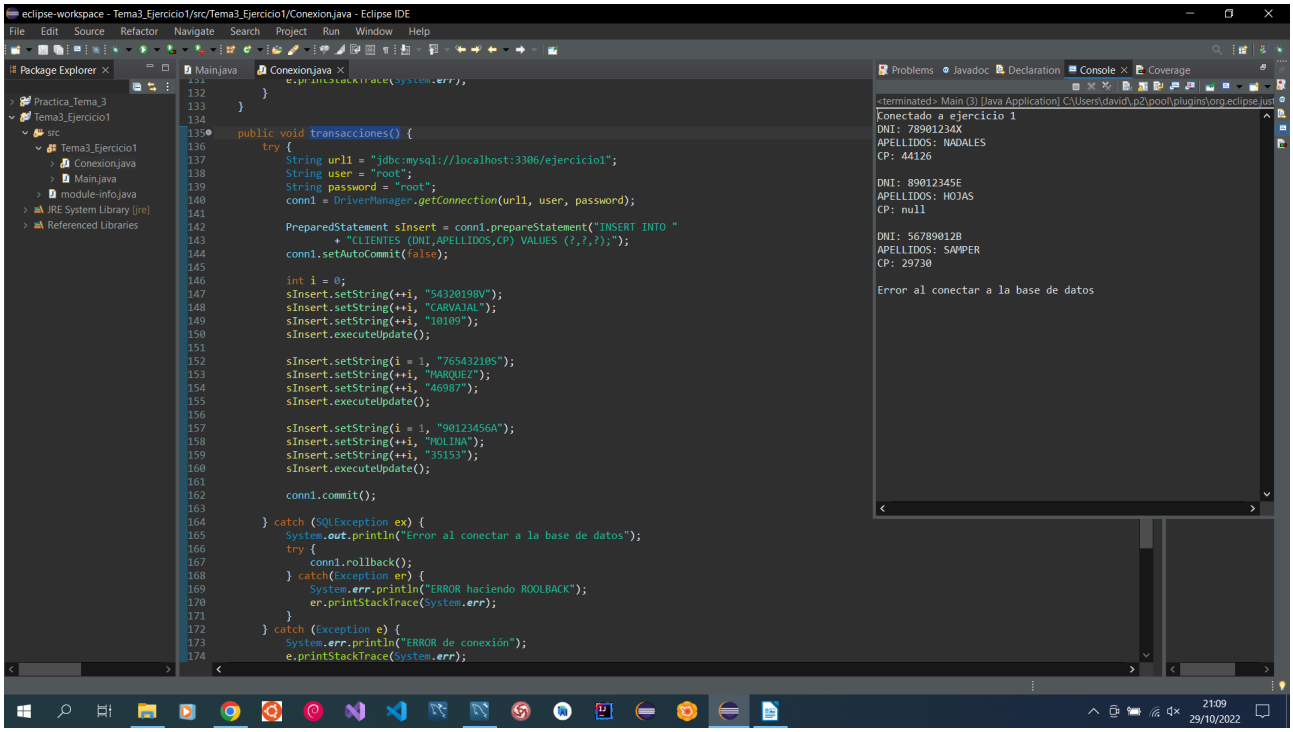


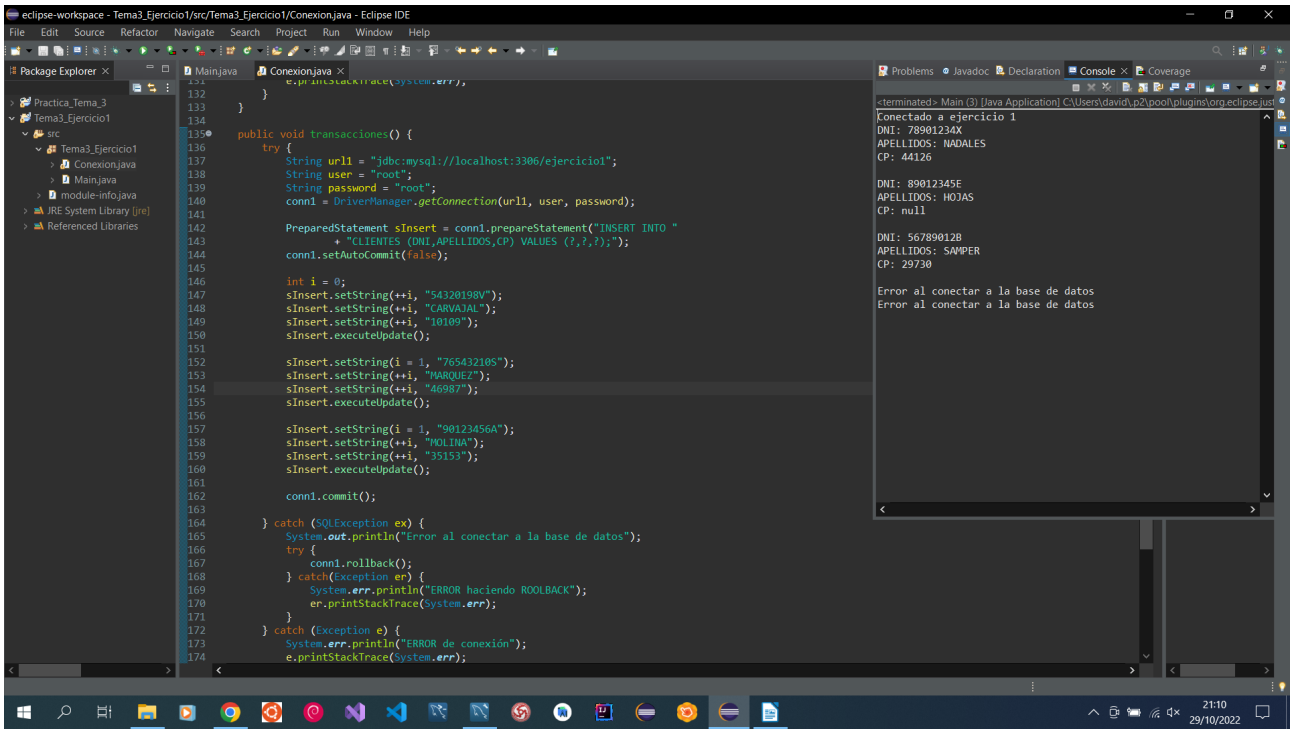

eclipse-workspace - Tema3_Ejercicio1/src/Tema3_Ejercicio1/Conexion.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Practica_Tema_3
Tema3_Ejercicio1
src
Tema3_Ejercicio1
Conexion.java
Main.java
module-info.java
JRE System Library [jre]
Referenced Libraries
Main.java
Conexion.java
Outline
Tema3_Ejercicio1
Conexion
conn1 : Connection
conexion() : void
cerrarconexion() : void
creartabla() : void
insertardatos() : void
consultasclien
Statement stm = conn1.createStatement();
ResultSet rs = stm.executeQuery("SELECT * FROM CLIENTES WHERE DNI='78901234X'");
int i = 1;
while (rs.next()) {
System.out.println("DNI: " + rs.getString("DNI"));
System.out.println("APELLIDOS: " + rs.getString("APELLIDOS"));
System.out.println("CP: " + rs.getString("CP"));
System.out.println();
}
rs = stm.executeQuery("SELECT * FROM CLIENTES WHERE DNI='89012345E'");
i = 1;
while (rs.next()) {
System.out.println("DNI: " + rs.getString("DNI"));
System.out.println("APELLIDOS: " + rs.getString("APELLIDOS"));
System.out.println("CP: " + rs.getString("CP"));
System.out.println();
}
rs = stm.executeQuery("SELECT * FROM CLIENTES WHERE DNI='56789012B'");
i = 1;
while (rs.next()) {
System.out.println("DNI: " + rs.getString("DNI"));
System.out.println("APELLIDOS: " + rs.getString("APELLIDOS"));
System.out.println("CP: " + rs.getString("CP"));
System.out.println();
}
} catch (SQLException ex) {
System.out.println("Error al conectar a la base de datos");
} catch (Exception e) {
e.printStackTrace(System.err);
Problems Javadoc Declaration Console Coverage
<terminated> Main (3) [Java Application] C:\Users\david\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20221004-1257\jre\bin\javaw.exe (29 oct 2022 20:30:37 – 20:30:37) [pid: 21652]
Conectado a ejercicio 1
DNI: 78901234X
APELLIDOS: NADALES
CP: 44126
DNI: 89012345E
APELLIDOS: HOJAS
CP: null
Writable
Overwrite
110 : 79 : 3392
20:31
29/10/2022

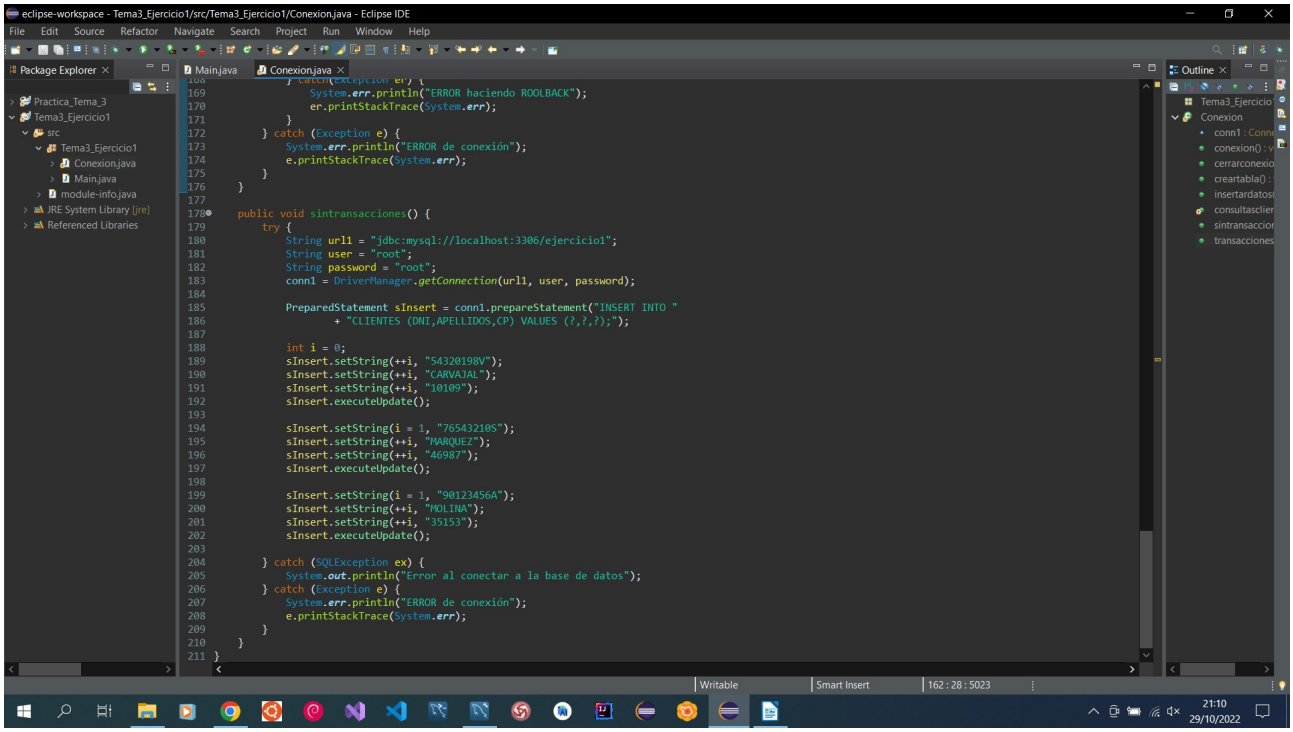

eclipse-workspace - Tema3_Ejercicio1/src/Tema3_Ejercicio1/Conexion.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Practica_Tema_3
Tema3_Ejercicio1
src
Tema3_Ejercicio1
Conexion.java
Main.java
module-info.java
JRE System Library [jre]
Referenced Libraries
Main.java
Conexion.java
Outline
Tema3_Ejercicio1
Conexion
conn1 : Connection
conexion() : void
cerrarconexion() : void
creartabla() : void
insertardatos() : void
consultasclien
int i = 1;
while (rs.next()) {
System.out.println("DNI: " + rs.getString("DNI"));
System.out.println("APELLIDOS: " + rs.getString("APELLIDOS"));
System.out.println("CP: " + rs.getString("CP"));
System.out.println();
}
rs = stm.executeQuery("SELECT * FROM CLIENTES WHERE DNI='89012345E'");
i = 1;
while (rs.next()) {
System.out.println("DNI: " + rs.getString("DNI"));
System.out.println("APELLIDOS: " + rs.getString("APELLIDOS"));
System.out.println("CP: " + rs.getString("CP"));
System.out.println();
}
rs = stm.executeQuery("SELECT * FROM CLIENTES WHERE DNI='56789012B'");
i = 1;
while (rs.next()) {
System.out.println("DNI: " + rs.getString("DNI"));
System.out.println("APELLIDOS: " + rs.getString("APELLIDOS"));
System.out.println("CP: " + rs.getString("CP"));
System.out.println();
}
} catch (SQLException ex) {
System.out.println("Error al conectar a la base de datos");
} catch (Exception e) {
e.printStackTrace(System.err);
}
}
}
Problems Javadoc Declaration Console Coverage
<terminated> Main (3) [Java Application] C:\Users\david\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20221004-1257\jre\bin\javaw.exe (29 oct 2022 20:30:37 – 20:30:37) [pid: 21652]
Conectado a ejercicio 1
DNI: 78901234X
APELLIDOS: NADALES
CP: 44126
DNI: 89012345E
APELLIDOS: HOJAS
CP: null
Writable
Overwrite
110 : 79 : 3392
20:31
29/10/2022

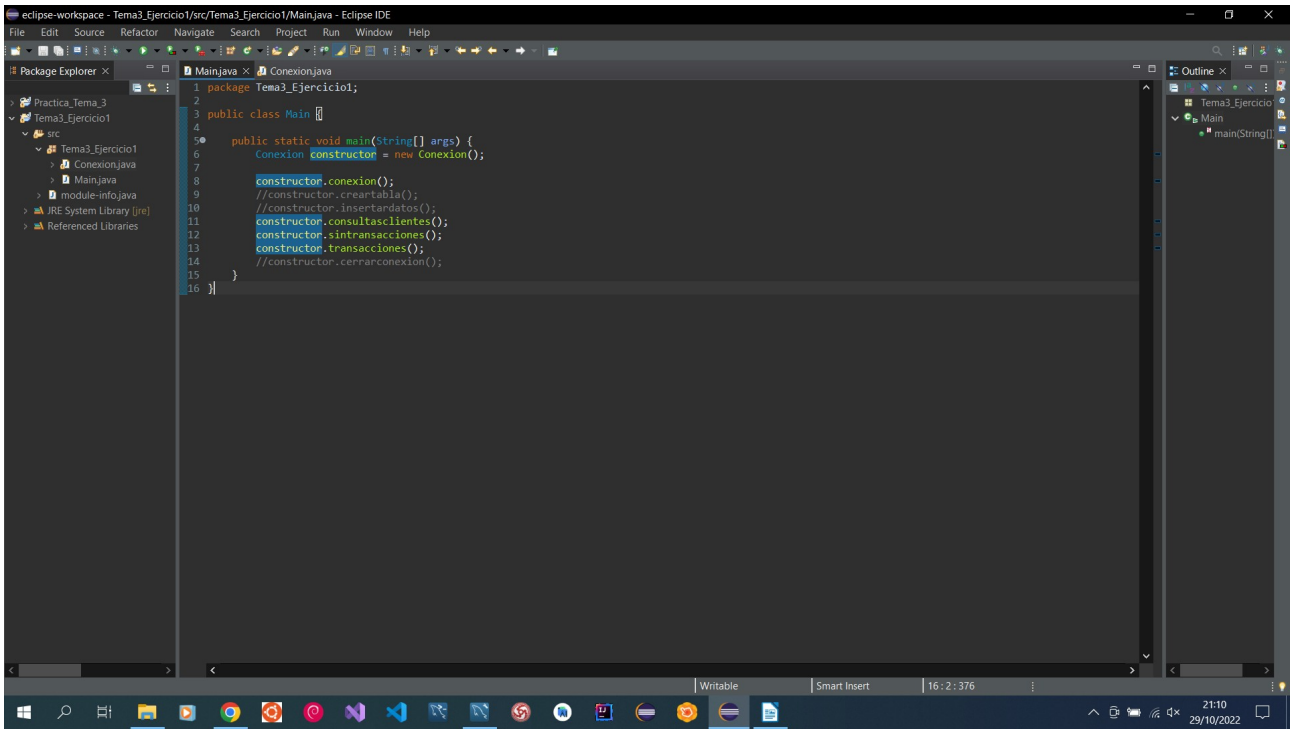
b) Comprueba las diferencias entre el programa anterior, en el que se agrupan todas la inserciones de registros con transacciones, y otro programa igual pero sin transacciones. Primero hay que hacer una copia del programa y eliminar el código que gestiona las transacciones (setAutoCommit, commit y rollback). Después se trata de probar ambos programas con un mismo conjunto de datos inicial en la tabla CLIENTES1, para comprobar la manera distinta en que se comportan. El conjunto de datos inicial y el programa podrían ser tales que las dos primeras inserciones se realizaran sin problemas, pero la última no, por haber ya en la tabla un cliente con ese DNI. El programa con transacciones no insertaría ningún registro. El programa sin transacciones insertará registros hasta que se produjera un error. La creación del conjunto de datos inicial debe automatizarse. Se puede hacer mediante una secuencia de sentencias SQL, que se pueden guardar en un fichero de texto, y ejecutar con el intérprete de SQL, o mediante un programa. En esas secuencias podrían borrarse todos los contenidos de la tabla con una sentencia DELETE y añadirse varias filas con sentencias INSERT.

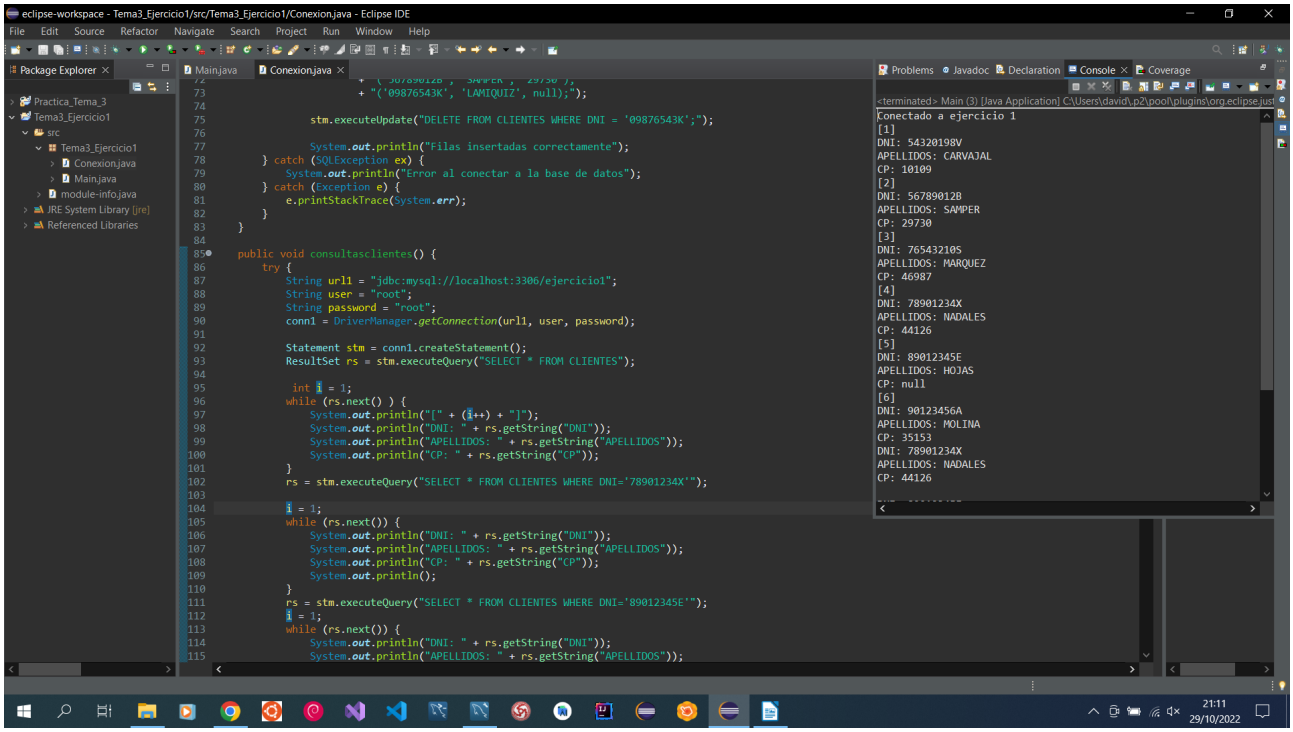



eclipse-workspace - Tema3_Ejercicio1/src/Tema3_Ejercicio1/Conexion.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Practica_Tema_3
Tema3_Ejercicio1
src
Tema3_Ejercicio1
Conexion.java
Main.java
module-info.java
JRE System Library [jre]
Referenced Libraries
Main.java
Conexion.java
public void transacciones() {
try {
String url1 = "jdbc:mysql://localhost:3306/ejercicio1";
String user = "root";
String password = "root";
conn1 = DriverManager.getConnection(url1, user, password);
PreparedStatement sInsert = conn1.prepareStatement("INSERT INTO "
+ "CLIENTES (DNI,APELLIDOS,CP) VALUES (?,?,?);");
conn1.setAutoCommit(false);
int i = 0;
sInsert.setString(++i, "54320198V");
sInsert.setString(++i, "CARVAJAL");
sInsert.setString(++i, "10109");
sInsert.executeUpdate();
sInsert.setString(i = 1, "76543210S");
sInsert.setString(++i, "MARQUEZ");
sInsert.setString(++i, "46987");
sInsert.executeUpdate();
sInsert.setString(i = 1, "90123456A");
sInsert.setString(++i, "MOLINA");
sInsert.setString(++i, "35153");
sInsert.executeUpdate();
conn1.commit();
} catch (SQLException ex) {
System.out.println("Error al conectar a la base de datos");
try {
conn1.rollback();
} catch(Exception er) {
System.err.println("ERROR haciendo ROOLBACK");
er.printStackTrace(System.err);
}
} catch (Exception e) {
System.err.println("ERROR de conexión");
e.printStackTrace(System.err);
Problems Javadoc Declaration Console Coverage
<terminated> Main (3) [Java Application] C:\Users\david\.p2\pool\plugins\org.eclipse.just
Conectado a ejercicio 1
DNI: 78901234X
APELLIDOS: NADALES
CP: 44126
DNI: 89012345E
APELLIDOS: HOJAS
CP: null
DNI: 56789012B
APELLIDOS: SAMPER
CP: 29730
Error al conectar a la base de datos
21:09
29/10/2022


eclipse-workspace - Tema3_Ejercicio1/src/Tema3_Ejercicio1/Conexion.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Practica_Tema_3
Tema3_Ejercicio1
src
Tema3_Ejercicio1
Conexion.java
Main.java
module-info.java
JRE System Library [jre]
Referenced Libraries
Main.java
Conexion.java
public void transacciones() {
try {
String url1 = "jdbc:mysql://localhost:3306/ejercicio1";
String user = "root";
String password = "root";
conn1 = DriverManager.getConnection(url1, user, password);
PreparedStatement sInsert = conn1.prepareStatement("INSERT INTO "
+ "CLIENTES (DNI,APELLIDOS,CP) VALUES (?,?,?);");
conn1.setAutoCommit(false);
int i = 0;
sInsert.setString(++i, "54320198V");
sInsert.setString(++i, "CARVAJAL");
sInsert.setString(++i, "10109");
sInsert.executeUpdate();
sInsert.setString(i = 1, "76543210S");
sInsert.setString(++i, "MARQUEZ");
sInsert.setString(++i, "46987");
sInsert.executeUpdate();
sInsert.setString(i = 1, "90123456A");
sInsert.setString(++i, "MOLINA");
sInsert.setString(++i, "35153");
sInsert.executeUpdate();
conn1.commit();
} catch (SQLException ex) {
System.out.println("Error al conectar a la base de datos");
try {
conn1.rollback();
} catch(Exception er) {
System.err.println("ERROR haciendo ROOLBACK");
er.printStackTrace(System.err);
}
} catch (Exception e) {
System.err.println("ERROR de conexión");
e.printStackTrace(System.err);
Problems Javadoc Declaration Console Coverage
<terminated> Main (3) [Java Application] C:\Users\david\.p2\pool\plugins\org.eclipse.just
Conectado a ejercicio 1
DNI: 78901234X
APELLIDOS: NADALES
CP: 44126
DNI: 89012345E
APELLIDOS: HOJAS
CP: null
DNI: 56789012B
APELLIDOS: SAMPER
CP: 29730
Error al conectar a la base de datos
Error al conectar a la base de datos
21:10
29/10/2022


eclipse-workspace - Tema3_Ejercicio1/src/Tema3_Ejercicio1/Conexion.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Main.java
Conexion.java
Outline
Practica_Tema_3
Tema3_Ejercicio1
src
Tema3_Ejercicio1
Conexion.java
Main.java
module-info.java
JRE System Library [jre]
Referenced Libraries
169 System.err.println("ERROR haciendo ROOLBACK");
170 er.printStackTrace(System.err);
171 }
172 } catch (Exception e) {
173 System.err.println("ERROR de conexión");
174 e.printStackTrace(System.err);
175 }
176 }
177
178 public void sintransacciones() {
179 try {
180 String url1 = "jdbc:mysql://localhost:3306/ejercicio1";
181 String user = "root";
182 String password = "root";
183 conn1 = DriverManager.getConnection(url1, user, password);
184
185 PreparedStatement sInsert = conn1.prepareStatement("INSERT INTO "
186 + "CLIENTES (DNI,APELLIDOS,CP) VALUES (?,?,?);");
187
188 int i = 0;
189 sInsert.setString(++i, "54320198V");
190 sInsert.setString(++i, "CARVAJAL");
191 sInsert.setString(++i, "10109");
192 sInsert.executeUpdate();
193
194 sInsert.setString(i = 1, "76543210S");
195 sInsert.setString(++i, "MARQUEZ");
196 sInsert.setString(++i, "46987");
197 sInsert.executeUpdate();
198
199 sInsert.setString(i = 1, "90123456A");
200 sInsert.setString(++i, "MOLINA");
201 sInsert.setString(++i, "35153");
202 sInsert.executeUpdate();
203
204 } catch (SQLException ex) {
205 System.out.println("Error al conectar a la base de datos");
206 } catch (Exception e) {
207 System.err.println("ERROR de conexión");
208 e.printStackTrace(System.err);
209 }
210 }
211 }
Writable
Smart Insert
162 : 28 : 5023
21:10
29/10/2022

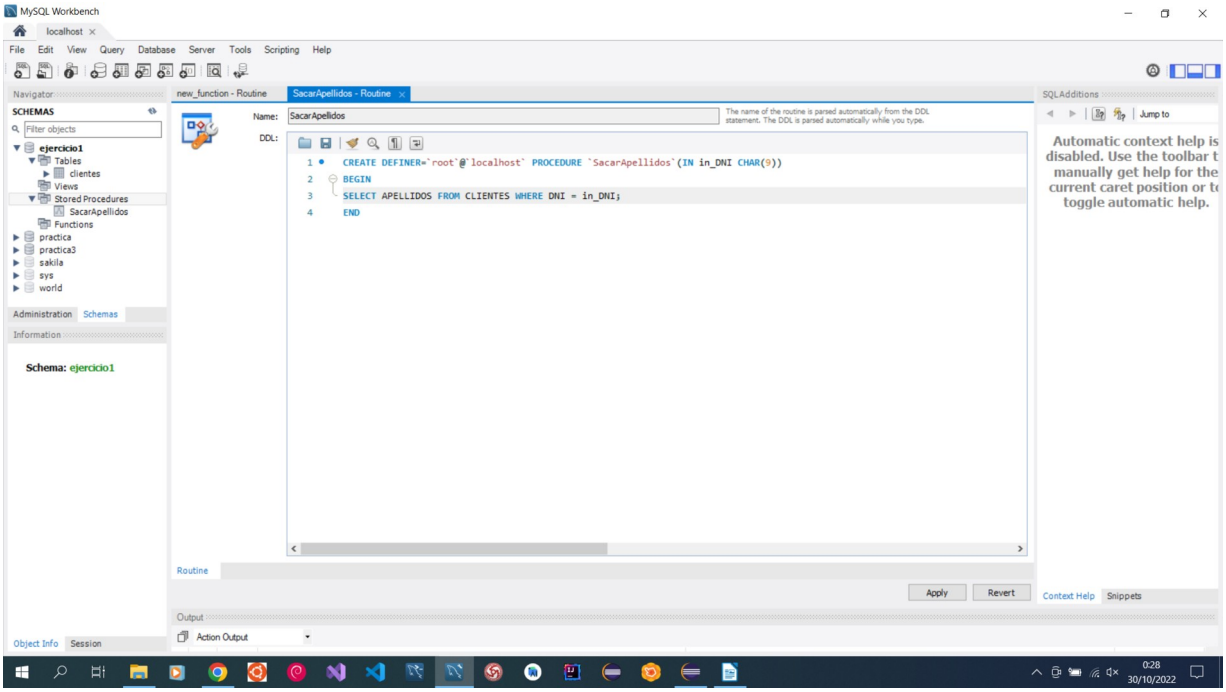

eclipse-workspace - Tema3_Ejercicio1/src/Tema3_Ejercicio1/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Main.java
Conexion.java
Outline
Practica_Tema_3
Tema3_Ejercicio1
src
Tema3_Ejercicio1
Conexion.java
Main.java
module-info.java
JRE System Library [jre]
Referenced Libraries
1 package Tema3_Ejercicio1;
2
3 public class Main {
4
5 public static void main(String[] args) {
6 Conexion constructor = new Conexion();
7
8 constructor.conexion();
9 //constructor.creartabla();
10 //constructor.insertardatos();
11 constructor.consultasclientes();
12 constructor.sintransacciones();
13 constructor.transacciones();
14 //constructor.cerrarconexion();
15 }
16 }
Writable
Smart Insert
16 : 2 : 376
21:10
29/10/2022

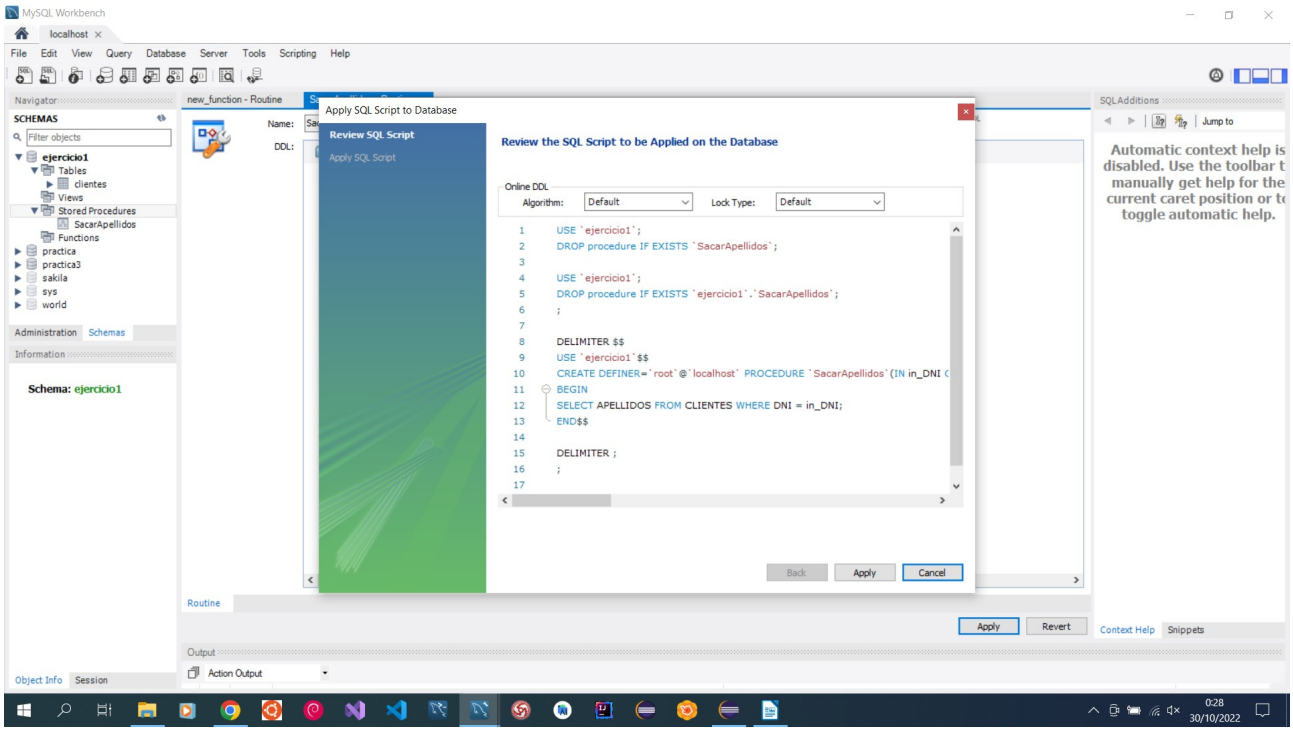


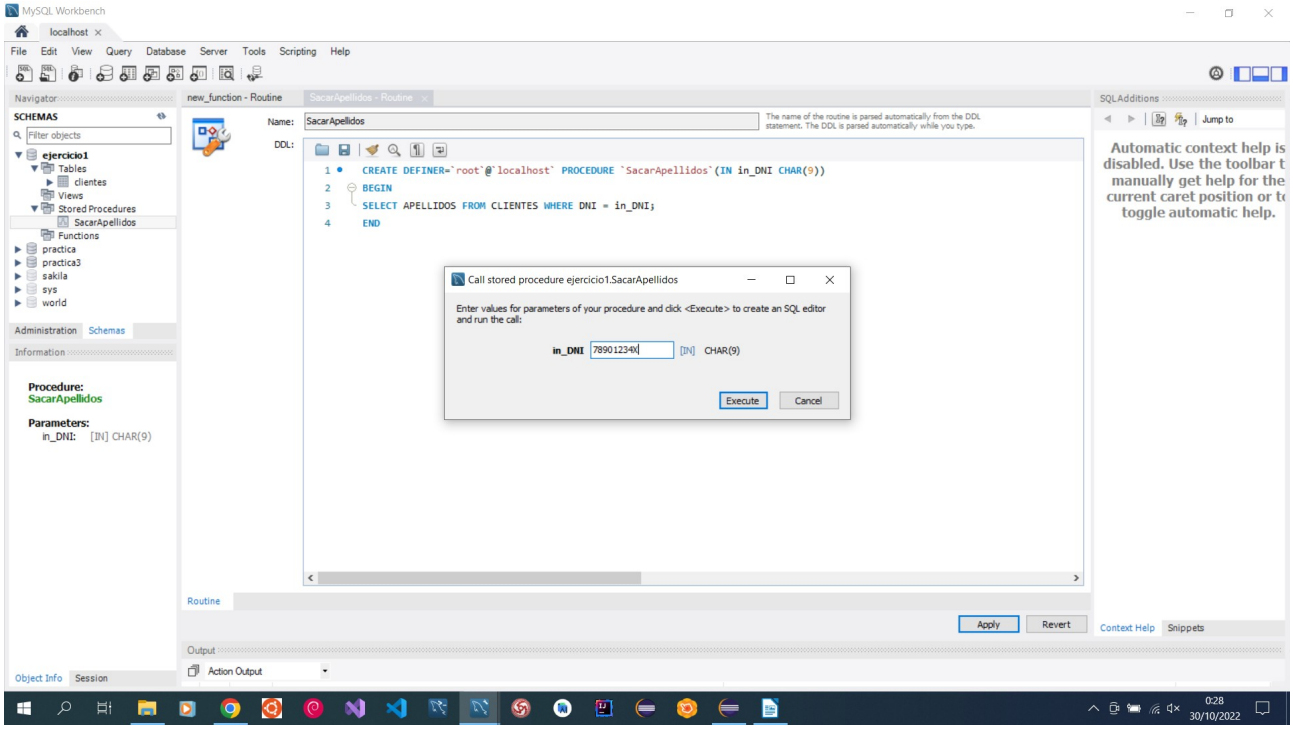
3.6 – PROCEDIMIENTOS Y FUNCIONES ALMACENADAS

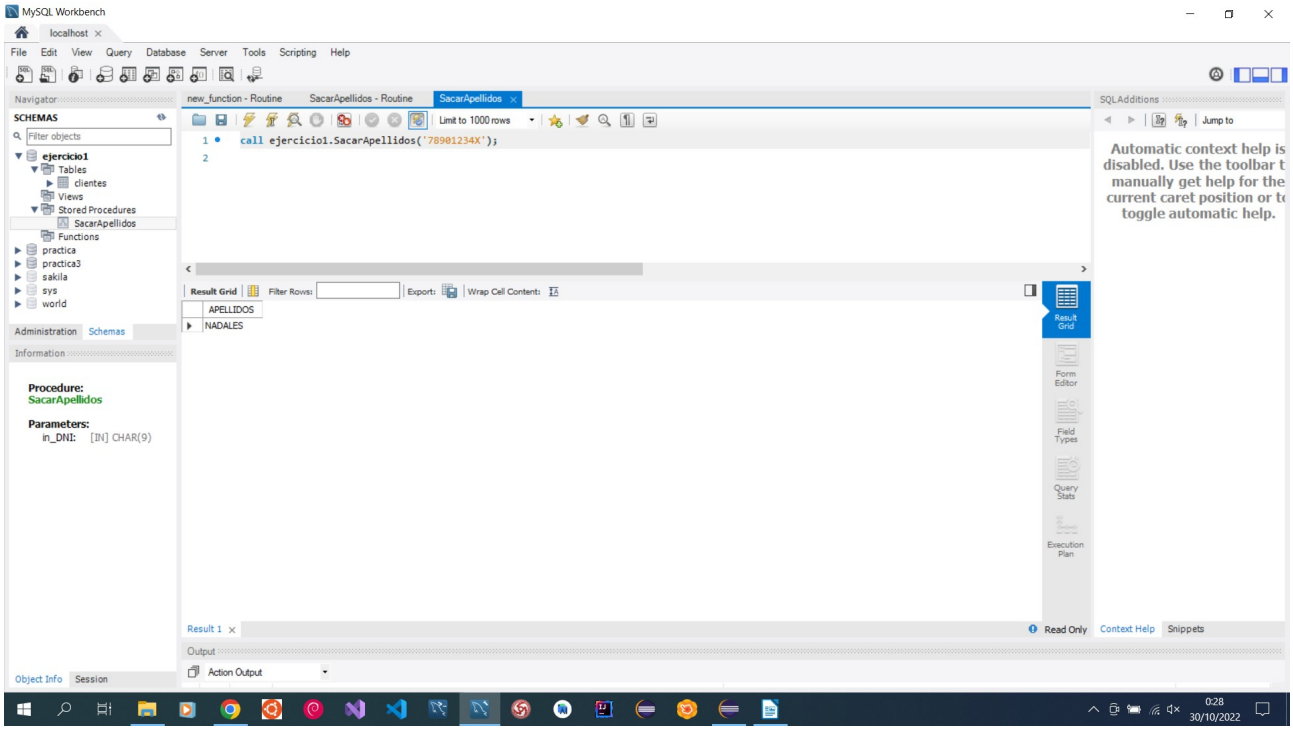
a) Crea una función almacenada con MySQL a la que se le pase el DNI de un cliente y que devuelva sus apellidos. Crea un programa en Java que realice una llamada a ella utilizando JDBC y escriba el valor devuelto por la función. Será necesario consultar documentación o investigar cómo crear funciones almacenadas en la base de datos utilizada, pero se hará de manera muy similar a un procedimiento almacenado.

- Creando un procedimiento:

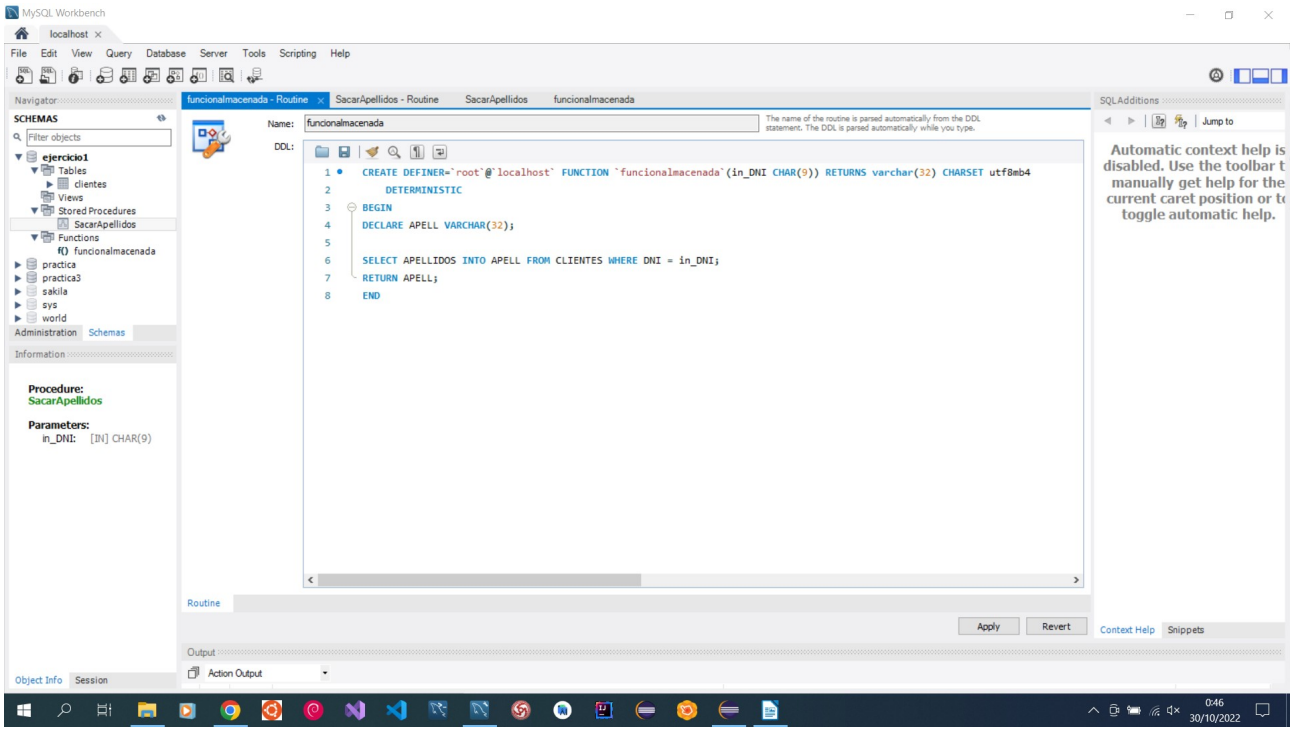


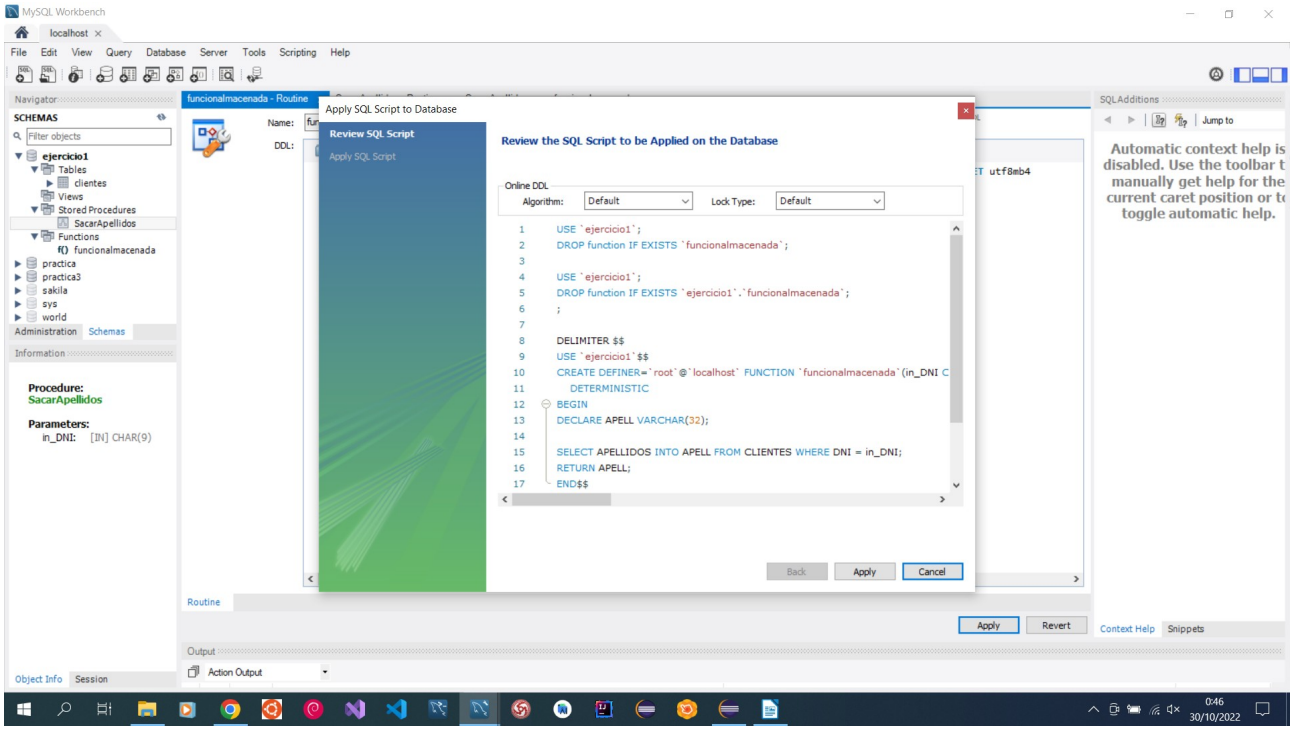

MySQL Workbench
Apply SQL Script to Database
Review SQL Script
Apply SQL Script
Review the SQL Script to be Applied on the Database
Online DDL
Algorithm: Default
Lock Type: Default
USE `ejercicio1`;
DROP procedure IF EXISTS `SacarApellidos`;
USE `ejercicio1`;
DROP procedure IF EXISTS `ejercicio1`.`SacarApellidos`;
;
DELIMITER $$
USE `ejercicio1`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `SacarApellidos`(IN in_DNI
BEGIN
SELECT APELLIDOS FROM CLIENTES WHERE DNI = in_DNI;
END$$
DELIMITER ;
;
Back
Apply
Cancel
Schema: ejercicio1

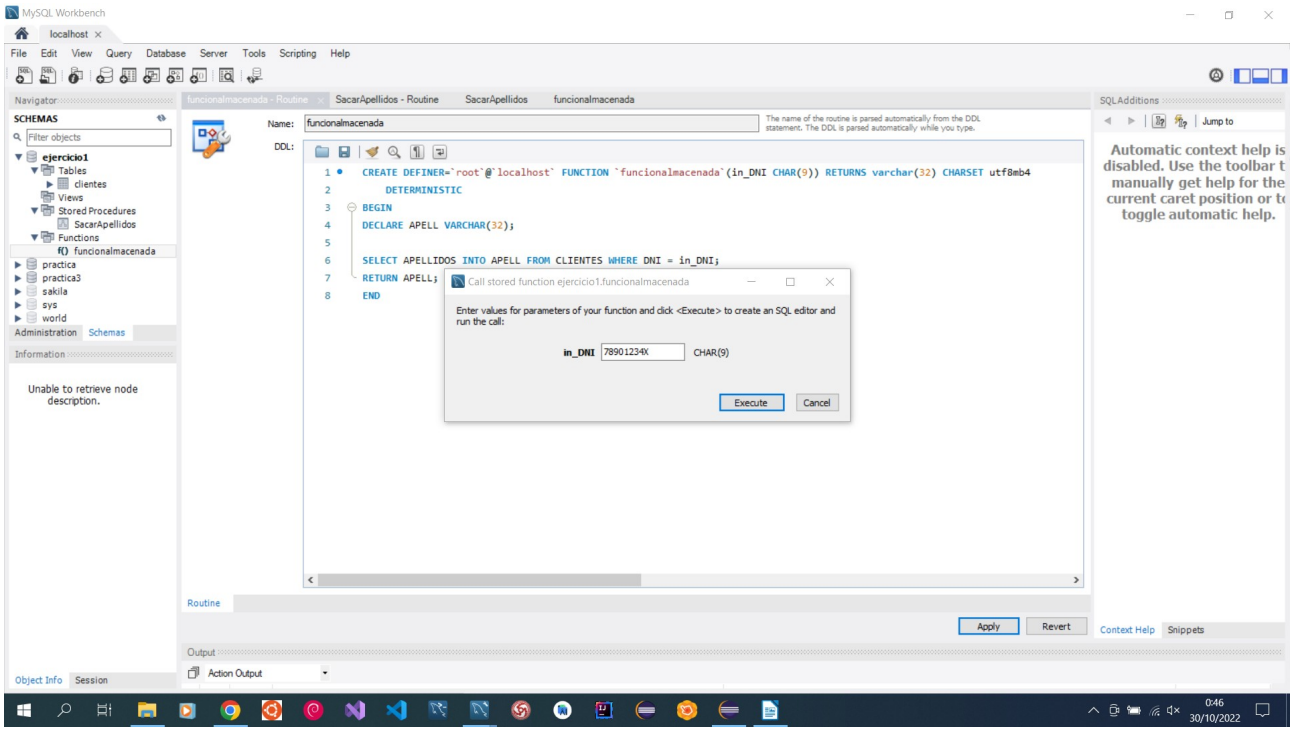

MySQL Workbench
CREATE DEFINER=`root`@`localhost` PROCEDURE `SacarApellidos`(IN in_DNI CHAR(9))
BEGIN
SELECT APELLIDOS FROM CLIENTES WHERE DNI = in_DNI;
END
Call stored procedure ejercicio1.SacarApellidos
Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:
in_DNI 78901234X [IN] CHAR(9)
Execute
Cancel
Procedure:
SacarApellidos
Parameters:
in_DNI: [IN] CHAR(9)

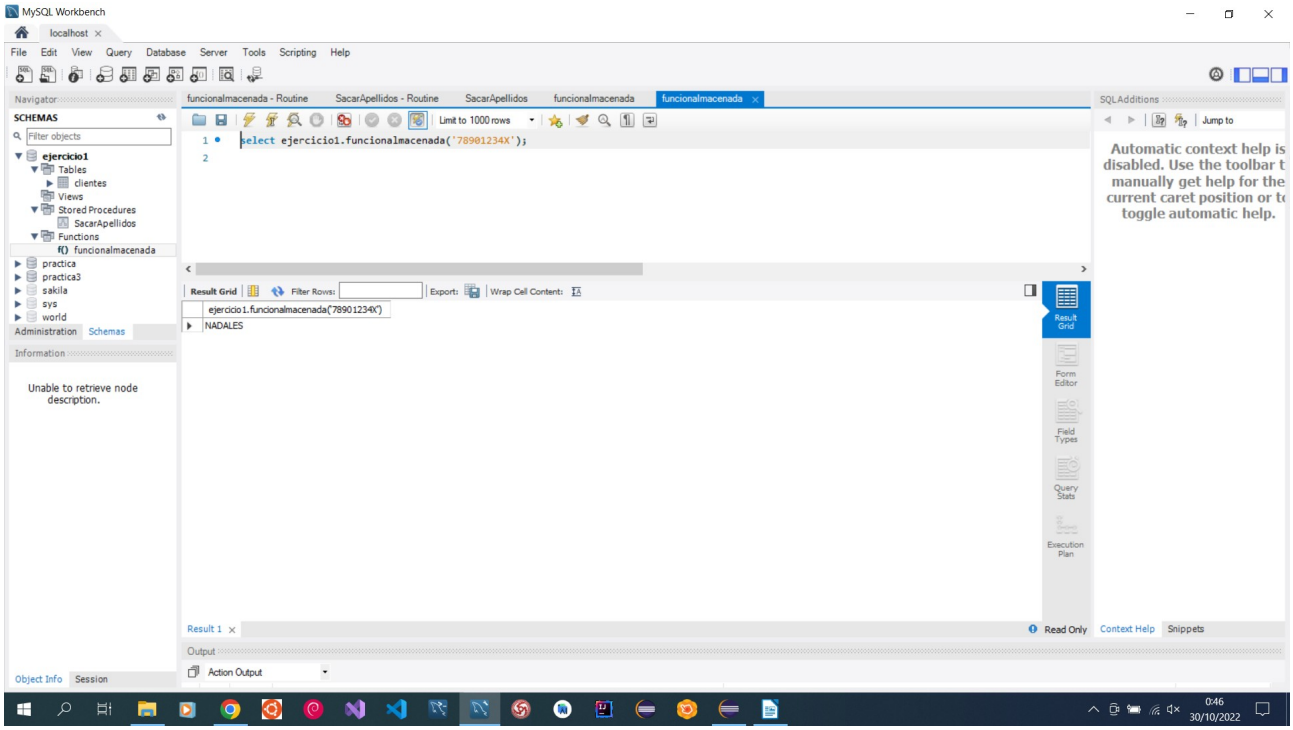


- Creando una función:

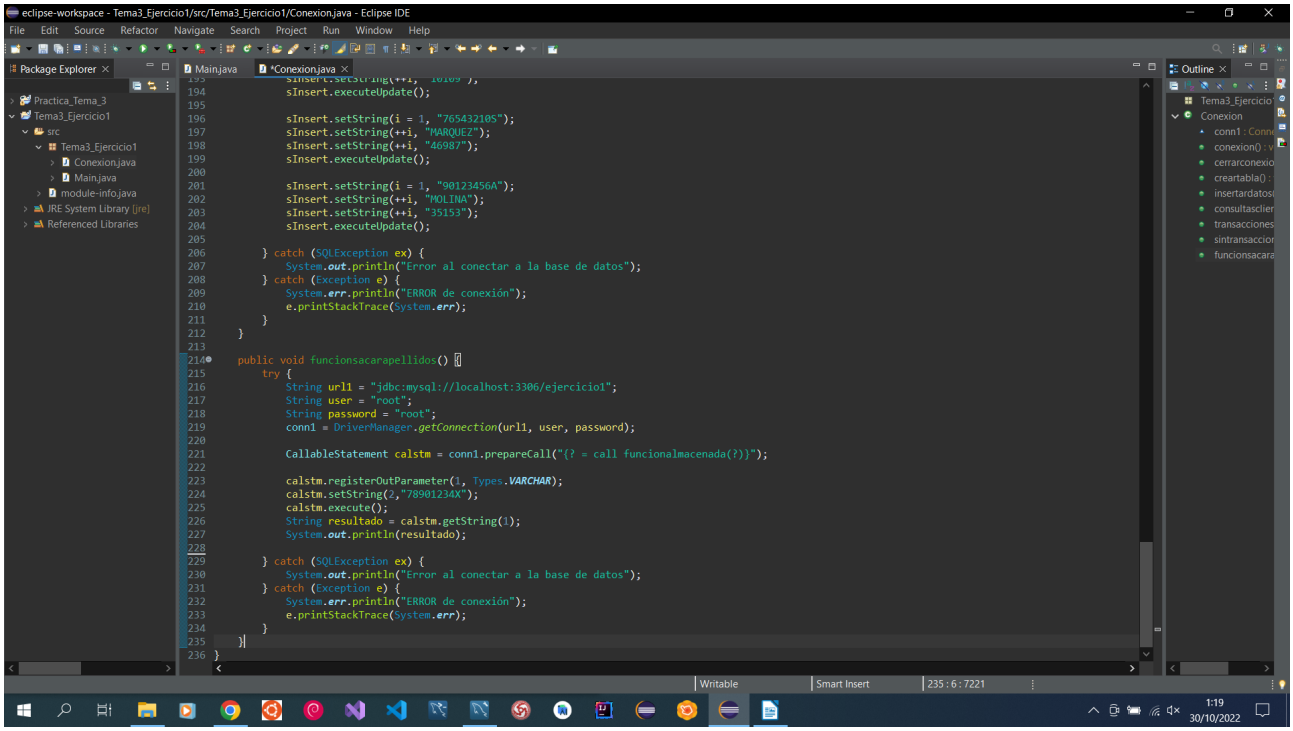


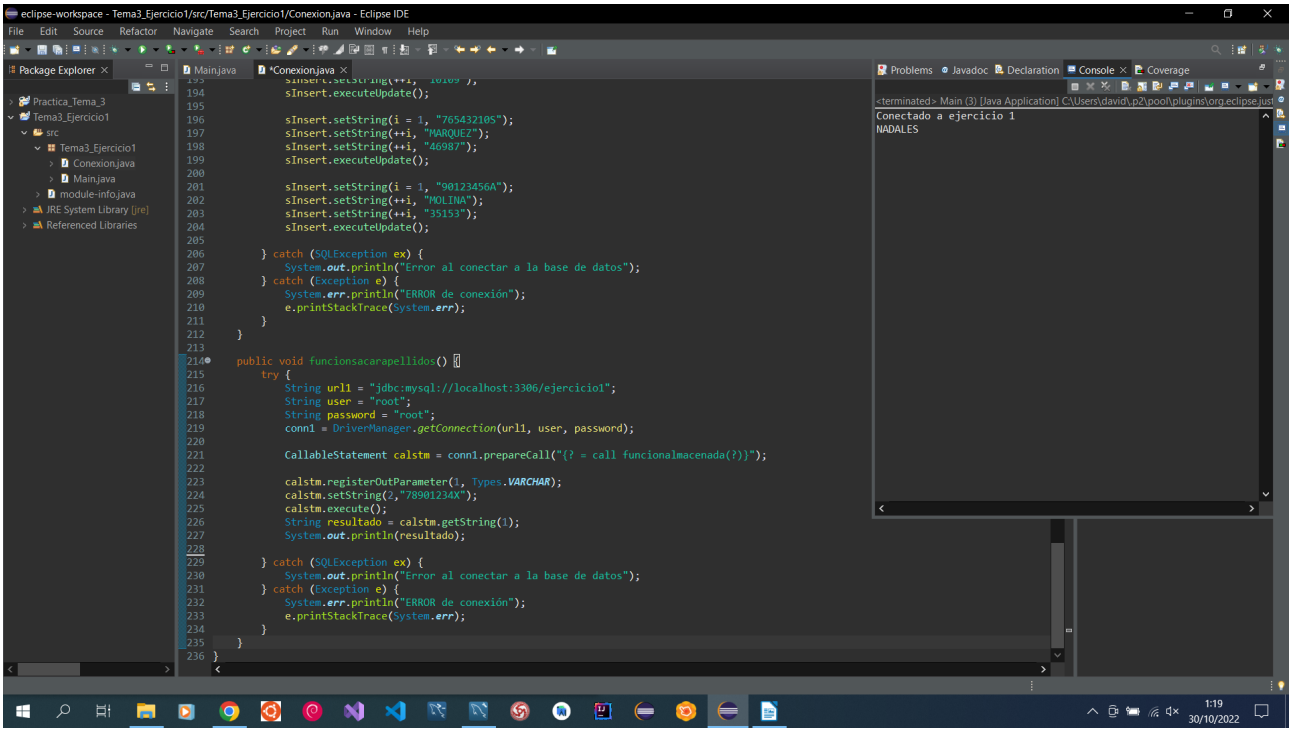

MySQL Workbench
localhost
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
ejercicio1
Tables
clientes
Views
Stored Procedures
SacarApellidos
Functions
f() funcionalmacenada
practica
practica3
sakila
sys
world
Administration Schemas
Information
Procedure:
SacarApellidos
Parameters:
in_DNI: [IN] CHAR(9)
funcionalmacenada - Routine
Name:
DDL:
Apply SQL Script to Database
Review SQL Script
Apply SQL Script
Review the SQL Script to be Applied on the Database
Online DDL
Algorithm: Default
Lock Type: Default
USE `ejercicio1`;
DROP function IF EXISTS `funcionalmacenada`;
USE `ejercicio1`;
DROP function IF EXISTS `ejercicio1`.`funcionalmacenada`;
;
DELIMITER $$
USE `ejercicio1`$$
CREATE DEFINER=`root`@`localhost` FUNCTION `funcionalmacenada`(in_DNI C
DETERMINISTIC
BEGIN
DECLARE APELL VARCHAR(32);
SELECT APELLIDOS INTO APELL FROM CLIENTES WHERE DNI = in_DNI;
RETURN APELL;
END$$
Back
Apply
Cancel
Routine
Apply
Revert
Output
Action Output
Object Info Session
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
0:46
30/10/2022


MySQL Workbench
localhost
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
ejercicio1
Tables
clientes
Views
Stored Procedures
SacarApellidos
Functions
f() funcionalmacenada
practica
practica3
sakila
sys
world
Administration Schemas
Information
Unable to retrieve node description.
funcionalmacenada - Routine
SacarApellidos - Routine
SacarApellidos
funcionalmacenada
Name: funcionalmacenada
The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.
DDL:
CREATE DEFINER=`root`@`localhost` FUNCTION `funcionalmacenada`(in_DNI CHAR(9)) RETURNS varchar(32) CHARSET utf8mb4
DETERMINISTIC
BEGIN
DECLARE APELL VARCHAR(32);
SELECT APELLIDOS INTO APELL FROM CLIENTES WHERE DNI = in_DNI;
RETURN APELL;
END
Call stored function ejercicio1.funcionalmacenada
Enter values for parameters of your function and click <Execute> to create an SQL editor and run the call:
in_DNI 78901234X CHAR(9)
Execute
Cancel
Routine
Apply
Revert
Output
Action Output
Object Info Session
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
0:46
30/10/2022

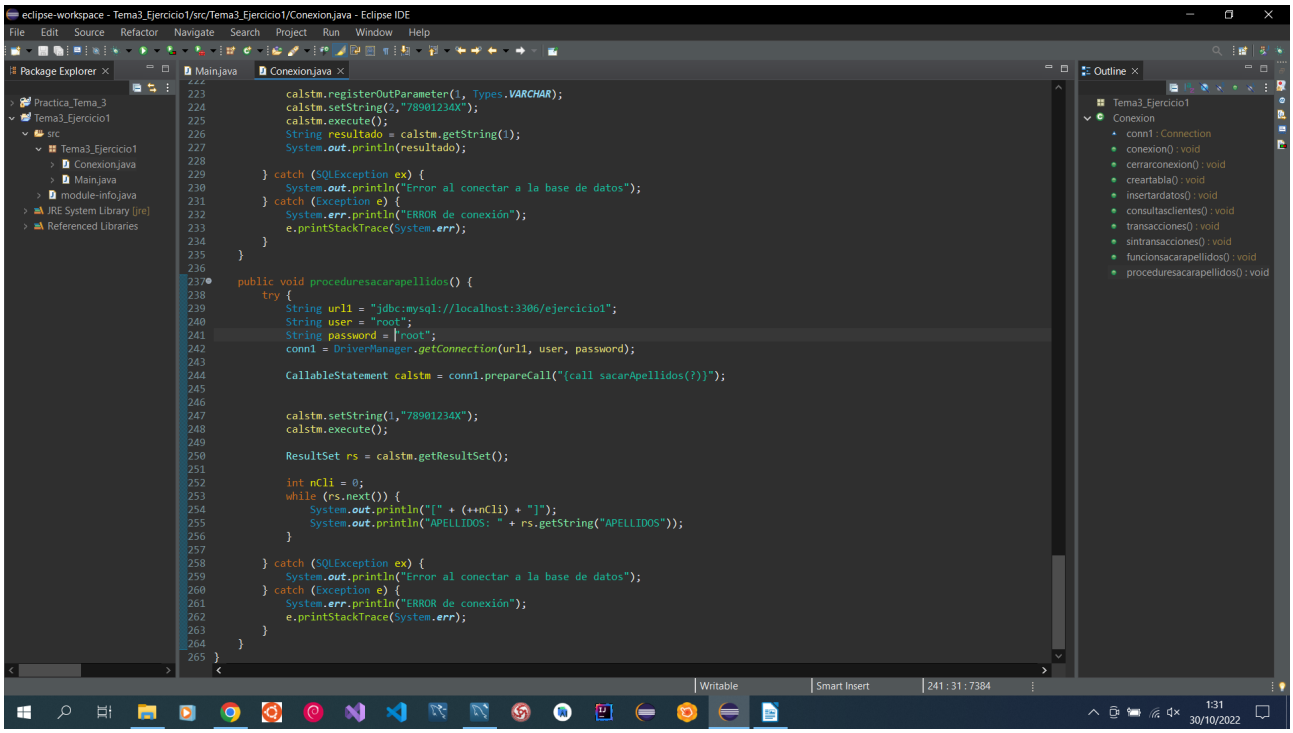


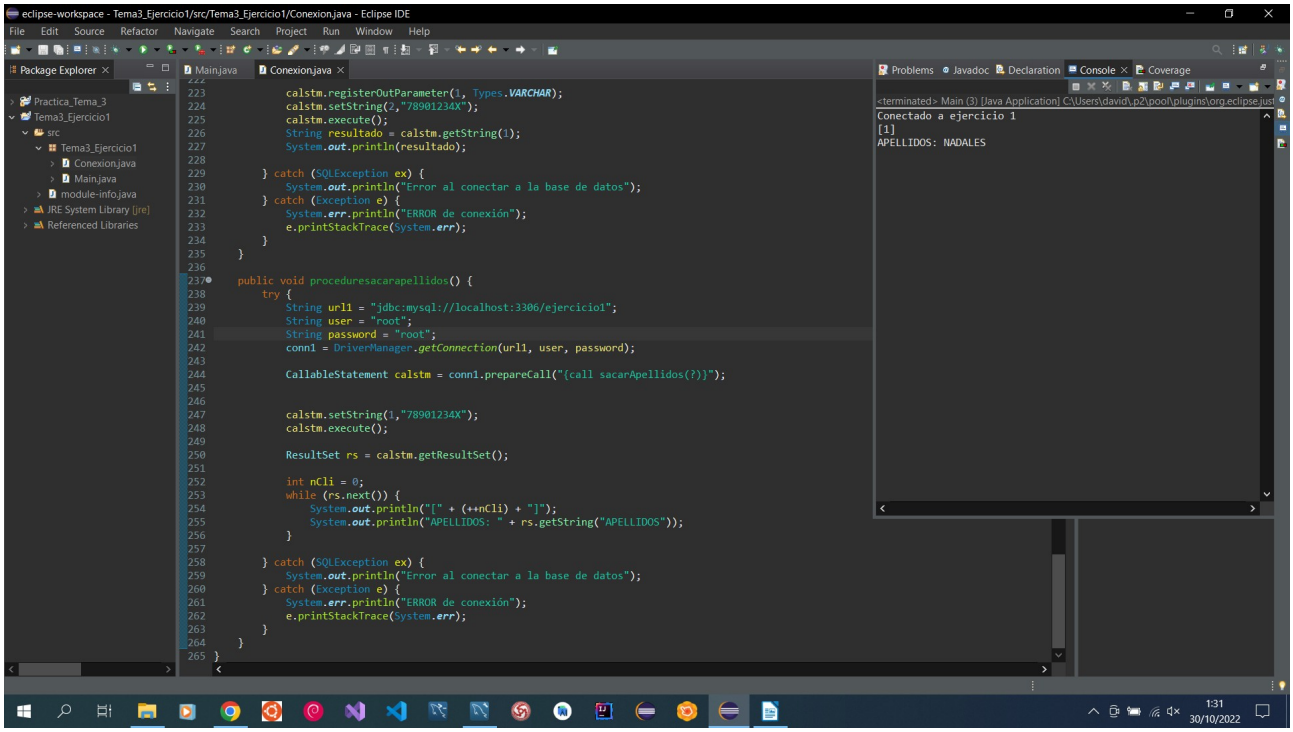
- Llamada a la función creada:





- Llamada al procedure creado:




eclipse-workspace - Tema3_Ejercicio1/src/Tema3_Ejercicio1/Conexion.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Practica_Tema_3
Tema3_Ejercicio1
src
Tema3_Ejercicio1
Conexion.java
Main.java
module-info.java
JRE System Library [jre]
Referenced Libraries
Main.java
Conexion.java
223 calstm.registerOutParameter(1, Types.VARCHAR);
224 calstm.setString(2,"78901234X");
225 calstm.execute();
226 String resultado = calstm.getString(1);
227 System.out.println(resultado);
228
229 } catch (SQLException ex) {
230 System.out.println("Error al conectar a la base de datos");
231 } catch (Exception e) {
232 System.err.println("ERROR de conexión");
233 e.printStackTrace(System.err);
234 }
235 }
236
237 public void proceduresacarapellidos() {
238 try {
239 String url1 = "jdbc:mysql://localhost:3306/ejercicio1";
240 String user = "root";
241 String password = "root";
242 conn1 = DriverManager.getConnection(url1, user, password);
243
244 CallableStatement calstm = conn1.prepareCall("{call sacarApellidos(?)}");
245
246
247 calstm.setString(1,"78901234X");
248 calstm.execute();
249
250 ResultSet rs = calstm.getResultSet();
251
252 int nCli = 0;
253 while (rs.next()) {
254 System.out.println("[" + (++nCli) + "]");
255 System.out.println("APELLIDOS: " + rs.getString("APELLIDOS"));
256 }
257
258 } catch (SQLException ex) {
259 System.out.println("Error al conectar a la base de datos");
260 } catch (Exception e) {
261 System.err.println("ERROR de conexión");
262 e.printStackTrace(System.err);
263 }
264 }
265 }
Problems Javadoc Declaration Console Coverage
<terminated> Main (3) [Java Application] C:\Users\david\.p2\pool\plugins\org.eclipse.just
Conectado a ejercicio 1
[1]
APELLIDOS: NADALES
1:31
30/10/2022