

The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows a project named 'Java\_PLSQL\_Net\_30' with a source folder 'src' containing a package 'com.oracle.net30package' and a class 'Java\_PLSQL\_Net\_30.java'. The main editor window shows the code for 'Java\_PLSQL\_Net\_30.java'. The code imports necessary JDBC and logging classes, establishes a connection to an Oracle database, and calls the 'NET\_30\_AVAIL\_CRED()' function. It then formats the result as a currency and prints it. The Outline view on the right shows the class structure. The bottom status bar indicates the application has terminated successfully.

```
1 package com.oracle.net30package;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.SQLException;
5 import java.sql.Types;
6 import java.sql.CallableStatement;
7 import java.text.NumberFormat;
8 import org.slf4j.Logger;
9 import org.slf4j.LoggerFactory;
10
11 public class Java_PLSQL_Net_30 {
12
13     public static void main(String[] args) throws SQLException {
14
15         Logger logger = LoggerFactory.getLogger(Java_PLSQL_Net_30.class);
16
17         try {
18             Connection conn = DriverManager.getConnection("jdbc:oracle:thin:username/password@//localhost:1521/pdb");
19             CallableStatement cstmt = conn.prepareCall("{? = call NET_30_AVAIL_CRED()}");
20             cstmt.registerOutParameter(1, Types.INTEGER);
21             cstmt.setInt(1, 10000001);
22             cstmt.execute();
23
24             NumberFormat defaultFormat = NumberFormat.getCurrencyInstance();
25             System.out.println("Net 30 Available Credit for Account ID 10000001: " + defaultFormat.format(cstmt.getInt(1)));
26
27         } catch (SQLException e) {
28             logger.warn(e.getMessage(), e);
29         }
30 }
```

<terminated> Java\_PLSQL\_Net\_30 [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (Jul 19, 2021, 11:03:46 PM – 11:03:47 PM)  
Net 30 Available Credit for Account ID 10000001: \$10,000.00

The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows a project named 'Java\_PLSQL\_Order' with a source folder 'src' containing a package 'com.oracle.datapackage' and a class 'Java\_PLSQL\_Order.java'. The main editor shows the code for 'Java\_PLSQL\_Order.java'.

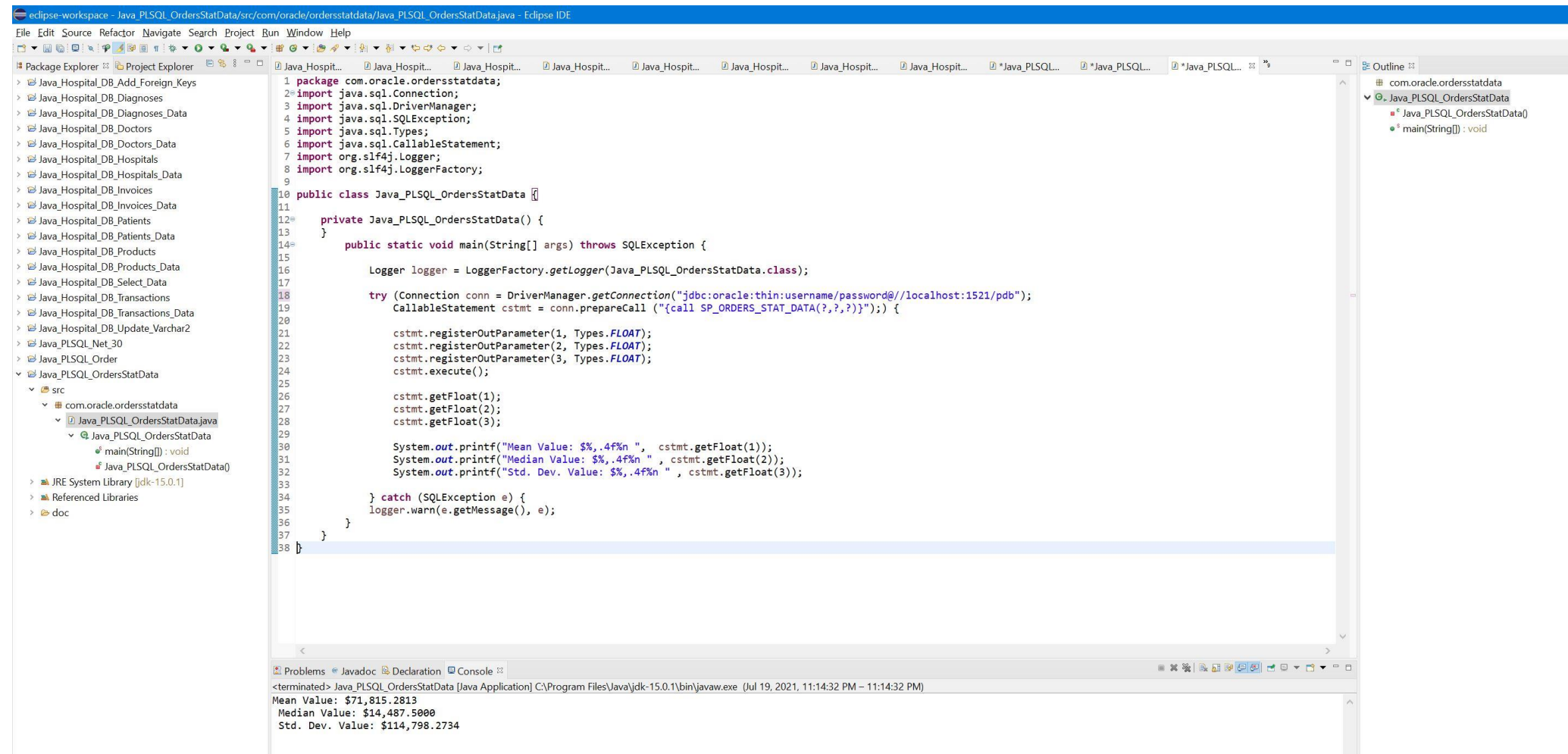
```
1 package com.oracle.datapackage;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.SQLException;
5 import java.sql.Statement;
6 import java.sql.ResultSet;
7 import org.slf4j.Logger;
8 import org.slf4j.LoggerFactory;
9
10 public class Java_PLSQL_Order {
11
12     public static void main(String[] args) throws SQLException {
13
14         String query = "SELECT * FROM username|JAVA_PL_SQL_ORD";
15         Logger logger = LoggerFactory.getLogger(Java_PLSQL_Order.class);
16
17         try (Connection conn = DriverManager.getConnection("jdbc:oracle:thin:username/password@//localhost:1521/pdb");
18             Statement stmt = conn.createStatement()) {
19             ResultSet rset = stmt.executeQuery(query);
20
21             while (rset.next())
22             {
23                 int javaOrdID = rset.getInt("JAVA_ORD_ID");
24                 int javaAcctID = rset.getInt("JAVA_ACCT_ID");
25                 int javaQuoteID = rset.getInt("JAVA_QUOTE_ID");
26                 int javaCatID = rset.getInt("JAVA_CAT_ID");
27
28                 System.out.print("JAVA_ORD_ID: " + javaOrdID);
29                 System.out.print(", JAVA_ACCT_ID: " + javaAcctID);
30                 System.out.print(", JAVA_QUOTE_ID: " + javaQuoteID);
31                 System.out.println(", JAVA_CAT_ID: " + javaCatID);
32             }
33             rset.close();
34
35         } catch (SQLException e) {
36             logger.warn(e.getMessage(), e);
37         }
38         System.out.println();
39         System.out.println("Connected to Oracle 19c | PL/SQL database successfully.");
40         System.out.println("No Oracle 19c, PL/SQL, Java, or connection errors to report.");
41     }
42 }
```

The Outline view on the right shows the package structure: 'com.oracle.datapackage' containing 'Java\_PLSQL\_Order' with a method 'main(String[]) : void'.

The Console view at the bottom shows the output of the application:

```
<terminated> Java_PLSQL_Order [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (Jul 19, 2021, 11:10:41 PM - 11:10:41 PM)
JAVA_ORD_ID: 1001, JAVA_ACCT_ID: 2001, JAVA_QUOTE_ID: 3001, JAVA_CAT_ID: 4001
JAVA_ORD_ID: 1002, JAVA_ACCT_ID: 2002, JAVA_QUOTE_ID: 3002, JAVA_CAT_ID: 4002
JAVA_ORD_ID: 1003, JAVA_ACCT_ID: 2003, JAVA_QUOTE_ID: 3003, JAVA_CAT_ID: 4003
JAVA_ORD_ID: 1004, JAVA_ACCT_ID: 2004, JAVA_QUOTE_ID: 3004, JAVA_CAT_ID: 4004
JAVA_ORD_ID: 1005, JAVA_ACCT_ID: 2005, JAVA_QUOTE_ID: 3005, JAVA_CAT_ID: 4005

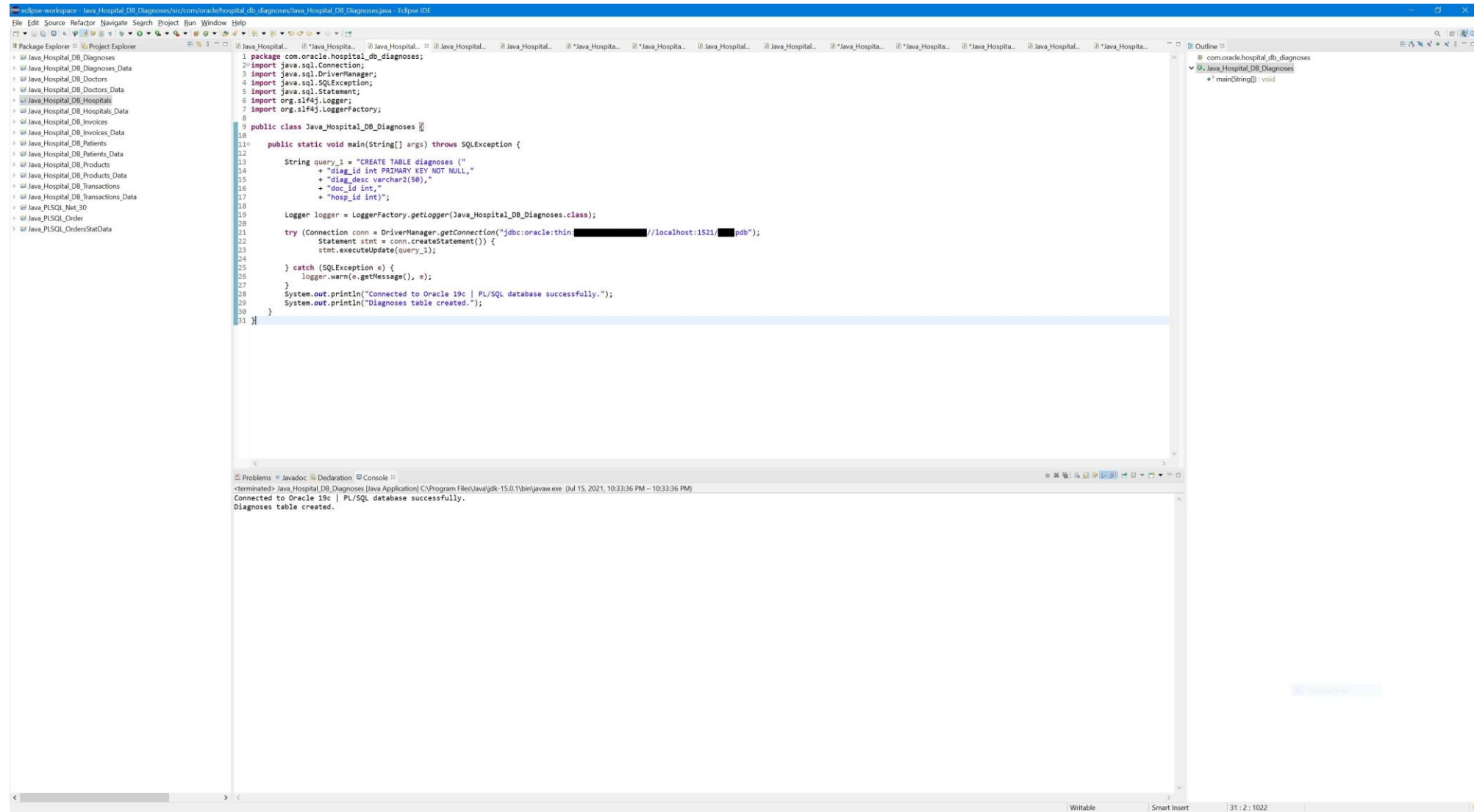
Connected to Oracle 19c | PL/SQL database successfully.
No Oracle 19c, PL/SQL, Java, or connection errors to report.
```



```
1 package com.oracle.ordersstatdata;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.SQLException;
5 import java.sql.Types;
6 import java.sql.CallableStatement;
7 import org.slf4j.Logger;
8 import org.slf4j.LoggerFactory;
9
10 public class Java_PLSQL_OrdersStatData {
11
12     private Java_PLSQL_OrdersStatData() {
13     }
14     public static void main(String[] args) throws SQLException {
15
16         Logger logger = LoggerFactory.getLogger(Java_PLSQL_OrdersStatData.class);
17
18         try (Connection conn = DriverManager.getConnection("jdbc:oracle:thin:username/password@//localhost:1521/pdb");
19             CallableStatement cstmt = conn.prepareCall("{call SP_ORDERS_STAT_DATA(?,?,?)}"); {
20
21             cstmt.registerOutParameter(1, Types.FLOAT);
22             cstmt.registerOutParameter(2, Types.FLOAT);
23             cstmt.registerOutParameter(3, Types.FLOAT);
24             cstmt.execute();
25
26             cstmt.getFloat(1);
27             cstmt.getFloat(2);
28             cstmt.getFloat(3);
29
30             System.out.printf("Mean Value: $%,.4f\n ", cstmt.getFloat(1));
31             System.out.printf("Median Value: $%,.4f\n ", cstmt.getFloat(2));
32             System.out.printf("Std. Dev. Value: $%,.4f\n ", cstmt.getFloat(3));
33
34         } catch (SQLException e) {
35             logger.warn(e.getMessage(), e);
36         }
37     }
38 }
```

com.oracle.ordersstatdata  
Java\_PLSQL\_OrdersStatData  
main(String[]): void

<terminated> Java\_PLSQL\_OrdersStatData [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (Jul 19, 2021, 11:14:32 PM – 11:14:32 PM)  
Mean Value: \$71,815.2813  
Median Value: \$14,487.5000  
Std. Dev. Value: \$114,798.2734

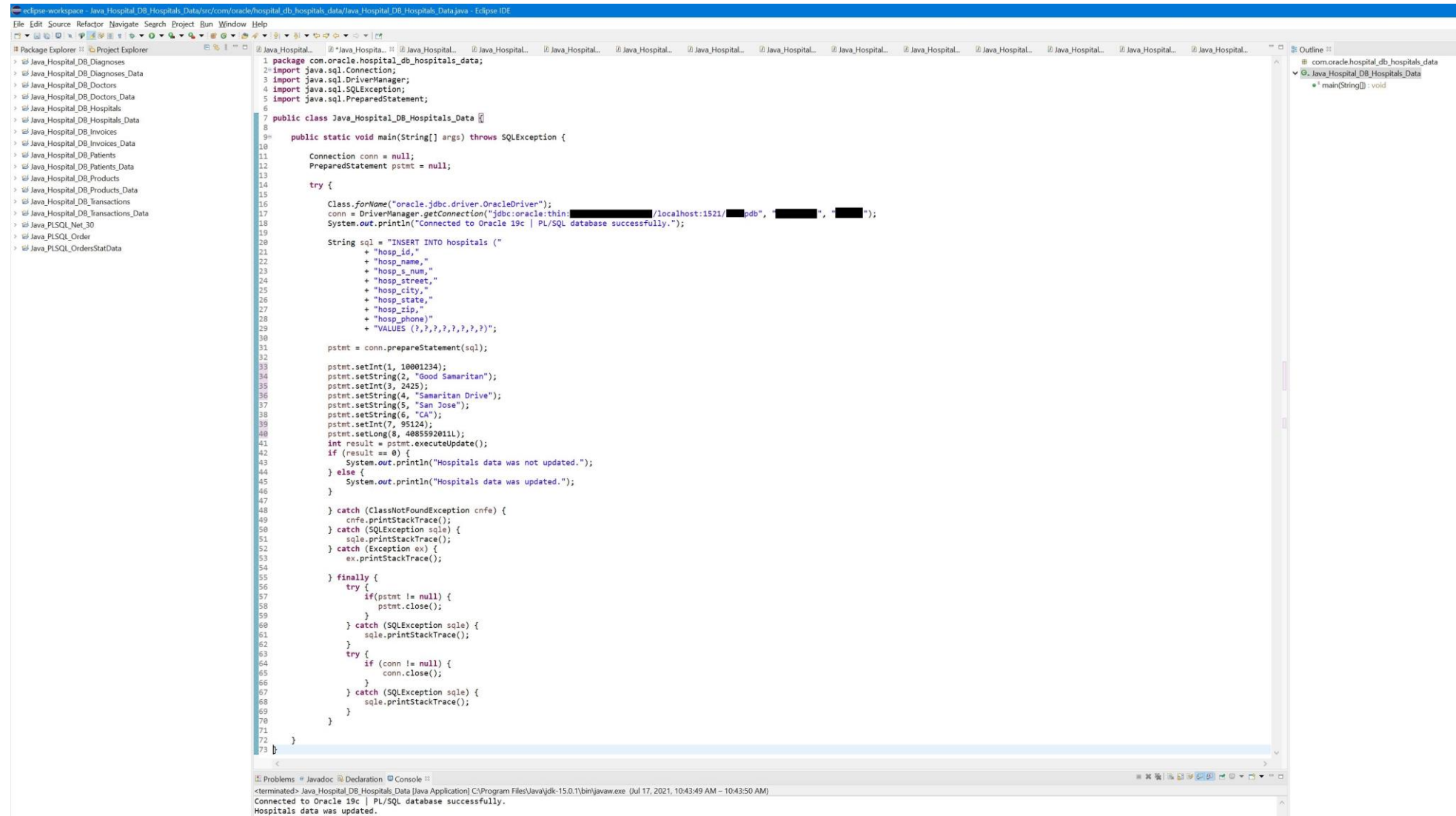


The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows a project named 'Java\_Hospital\_DB\_Diagnoses' with several sub-packages. The main editor window shows the following Java code:

```
1 package com.oracle.hospital_db_diagnoses;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.SQLException;
5 import java.sql.Statement;
6 import org.slf4j.Logger;
7 import org.slf4j.LoggerFactory;
8
9 public class Java_Hospital_DB_Diagnoses {
10
11     public static void main(String[] args) throws SQLException {
12
13         String query_1 = "CREATE TABLE diagnoses ("
14             + "diag_id int PRIMARY KEY NOT NULL,"
15             + "diag_desc Varchar2(50),"
16             + "doc_id int,"
17             + "hosp_id int)";
18
19         Logger logger = LoggerFactory.getLogger(Java_Hospital_DB_Diagnoses.class);
20
21         try (Connection conn = DriverManager.getConnection("jdbc:oracle:thin://localhost:1521/");
22             Statement stmt = conn.createStatement()) {
23             stmt.executeUpdate(query_1);
24         } catch (SQLException e) {
25             logger.warn(e.getMessage(), e);
26         }
27         System.out.println("Connected to Oracle 19c | PL/SQL database successfully.");
28         System.out.println("Diagnoses table created.");
29     }
30 }
31 }
```

The Outline view on the right shows the package structure, and the Console view at the bottom displays the execution output:

```
<terminated> Java_Hospital_DB_Diagnoses [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (Jul 15, 2021, 10:33:36 PM - 10:33:36 PM)
Connected to Oracle 19c | PL/SQL database successfully.
Diagnoses table created.
```

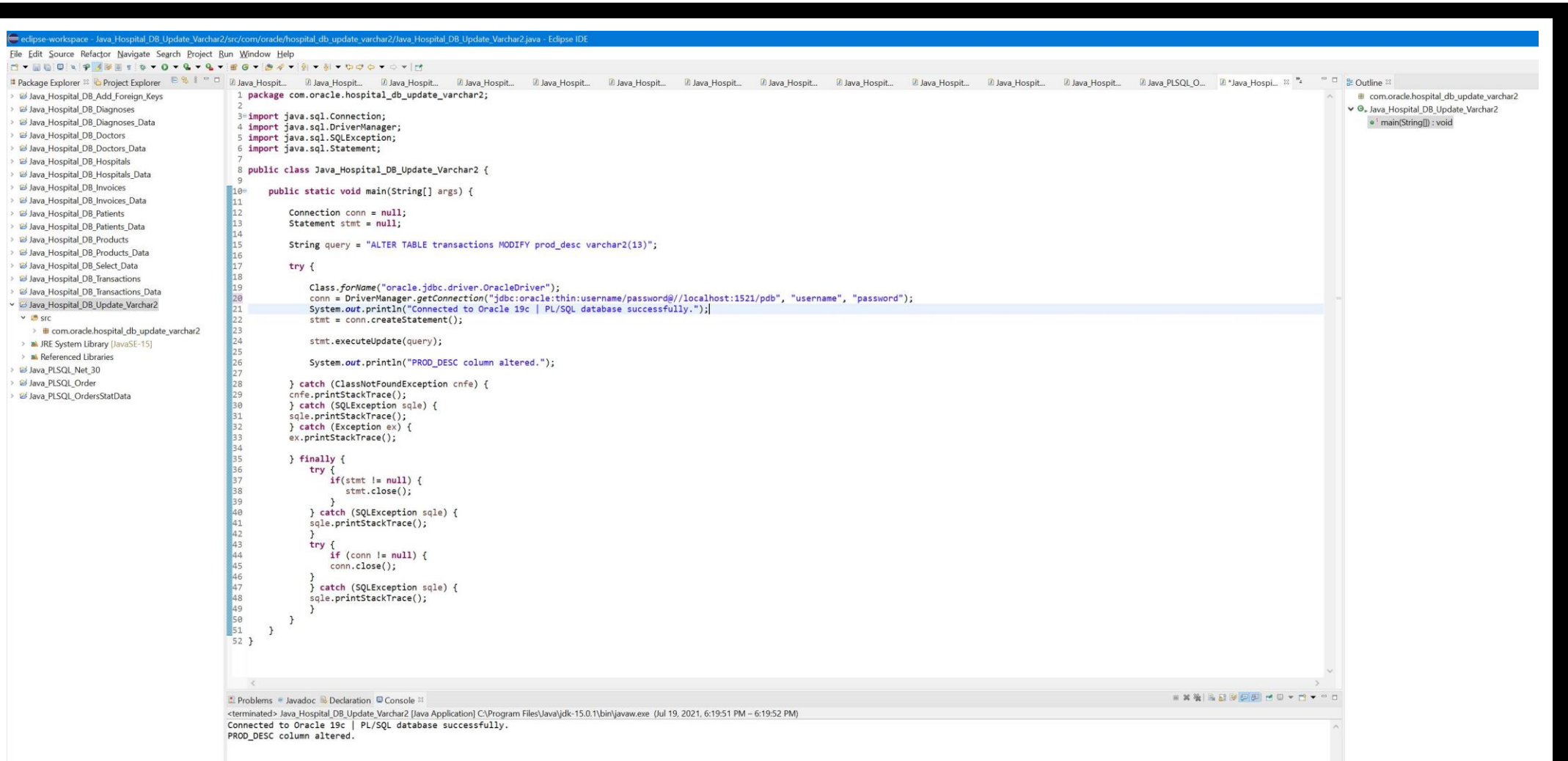


```
1 package com.oracle.hospital.db.hospitals_data;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.SQLException;
5 import java.sql.PreparedStatement;
6
7 public class Java_Hospital_DB_Hospitals_Data {
8
9     public static void main(String[] args) throws SQLException {
10
11         Connection conn = null;
12         PreparedStatement pstmt = null;
13
14         try {
15
16             Class.forName("oracle.jdbc.driver.OracleDriver");
17             conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521://pdb", " ", " ");
18             System.out.println("Connected to Oracle 19c | PL/SQL database successfully.");
19
20             String sql = "INSERT INTO hospitals ("
21                 + "hosp_id,"
22                 + "hosp_name,"
23                 + "hosp_s_num,"
24                 + "hosp_street,"
25                 + "hosp_city,"
26                 + "hosp_state,"
27                 + "hosp_zip,"
28                 + "hosp_phone)"
29                 + "VALUES (?, ?, ?, ?, ?, ?, ?)";
30
31             pstmt = conn.prepareStatement(sql);
32
33             pstmt.setInt(1, 10001234);
34             pstmt.setString(2, "Good Samaritan");
35             pstmt.setInt(3, 2425);
36             pstmt.setString(4, "Samaritan Drive");
37             pstmt.setString(5, "San Jose");
38             pstmt.setString(6, "CA");
39             pstmt.setInt(7, 95124);
40             pstmt.setLong(8, 4085592011L);
41             int result = pstmt.executeUpdate();
42             if (result == 0) {
43                 System.out.println("Hospitals data was not updated.");
44             } else {
45                 System.out.println("Hospitals data was updated.");
46             }
47
48             } catch (ClassNotFoundException cnfe) {
49                 cnfe.printStackTrace();
50             } catch (SQLException sqle) {
51                 sqle.printStackTrace();
52             } catch (Exception ex) {
53                 ex.printStackTrace();
54             }
55         } finally {
56             try {
57                 if (pstmt != null) {
58                     pstmt.close();
59                 }
60             } catch (SQLException sqle) {
61                 sqle.printStackTrace();
62             }
63             try {
64                 if (conn != null) {
65                     conn.close();
66                 }
67             } catch (SQLException sqle) {
68                 sqle.printStackTrace();
69             }
70         }
71     }
72 }
73 }
```

com.oracle.hospital.db.hospitals\_data  
Java\_Hospital\_DB\_Hospitals\_Data  
main(String[]) - void

Problems Javadoc Declaration Console  
<terminated> Java\_Hospital\_DB\_Hospitals\_Data [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (Jul 17, 2021, 10:43:49 AM - 10:43:50 AM)  
Connected to Oracle 19c | PL/SQL database successfully.  
Hospitals data was updated.

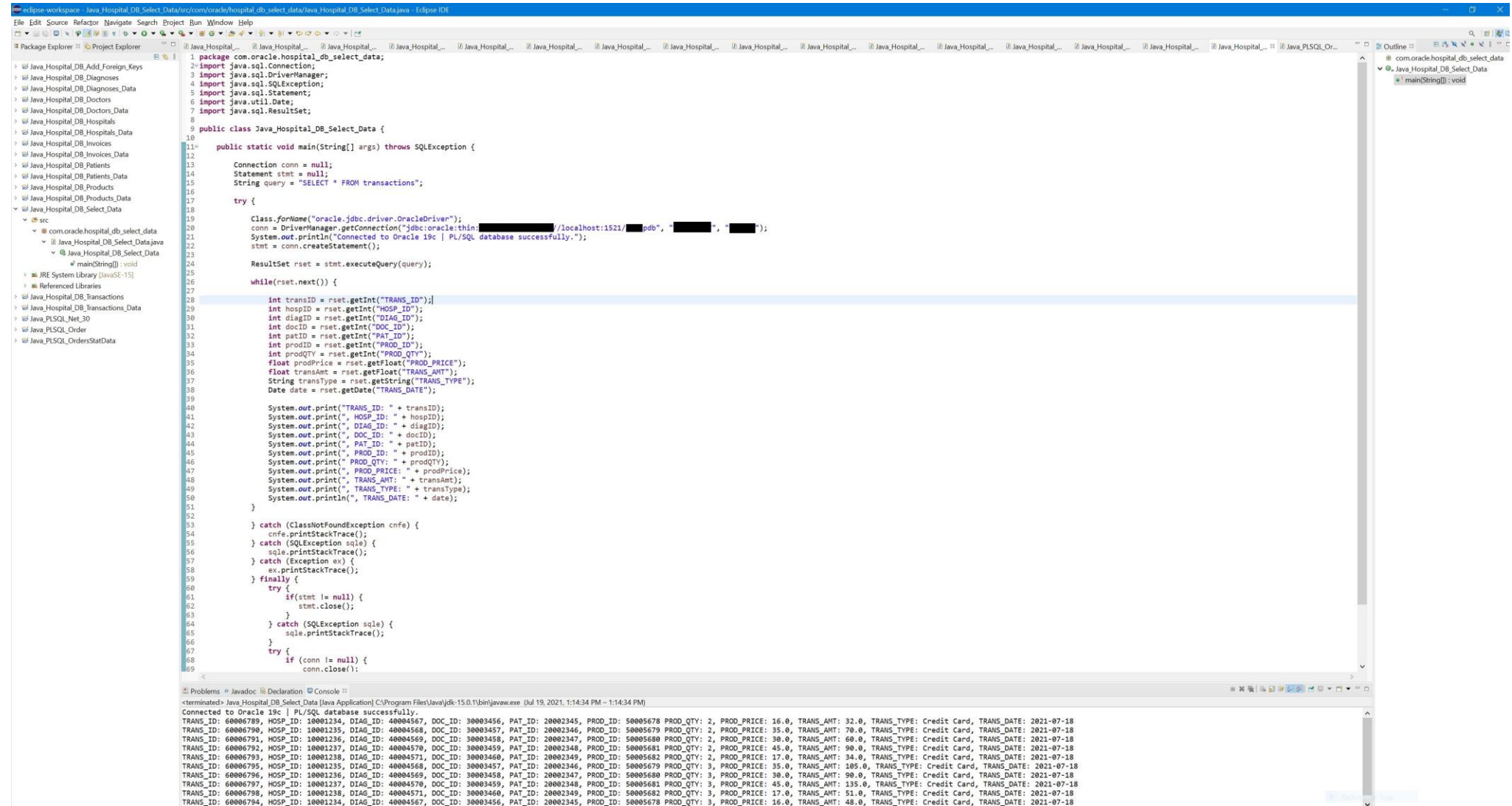




```
1 package com.oracle.hospital_db_update_varchar2;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.SQLException;
6 import java.sql.Statement;
7
8 public class Java_Hospital_DB_Update_Varchar2 {
9
10     public static void main(String[] args) {
11
12         Connection conn = null;
13         Statement stmt = null;
14
15         String query = "ALTER TABLE transactions MODIFY prod_desc varchar2(13)";
16
17         try {
18
19             Class.forName("oracle.jdbc.driver.OracleDriver");
20             conn = DriverManager.getConnection("jdbc:oracle:thin:username/password@//localhost:1521/pdb", "username", "password");
21             System.out.println("Connected to Oracle 19c | PL/SQL database successfully.");
22             stmt = conn.createStatement();
23
24             stmt.executeUpdate(query);
25
26             System.out.println("PROD_DESC column altered.");
27
28         } catch (ClassNotFoundException cnfe) {
29             cnfe.printStackTrace();
30         } catch (SQLException sqle) {
31             sqle.printStackTrace();
32         } catch (Exception ex) {
33             ex.printStackTrace();
34         }
35
36         finally {
37             try {
38                 if (stmt != null) {
39                     stmt.close();
40                 }
41             } catch (SQLException sqle) {
42                 sqle.printStackTrace();
43             }
44             try {
45                 if (conn != null) {
46                     conn.close();
47                 }
48             } catch (SQLException sqle) {
49                 sqle.printStackTrace();
50             }
51         }
52     }
53 }
```

com.oracle.hospital\_db\_update\_varchar2  
▼ Java\_Hospital\_DB\_Update\_Varchar2  
main(String[]) : void

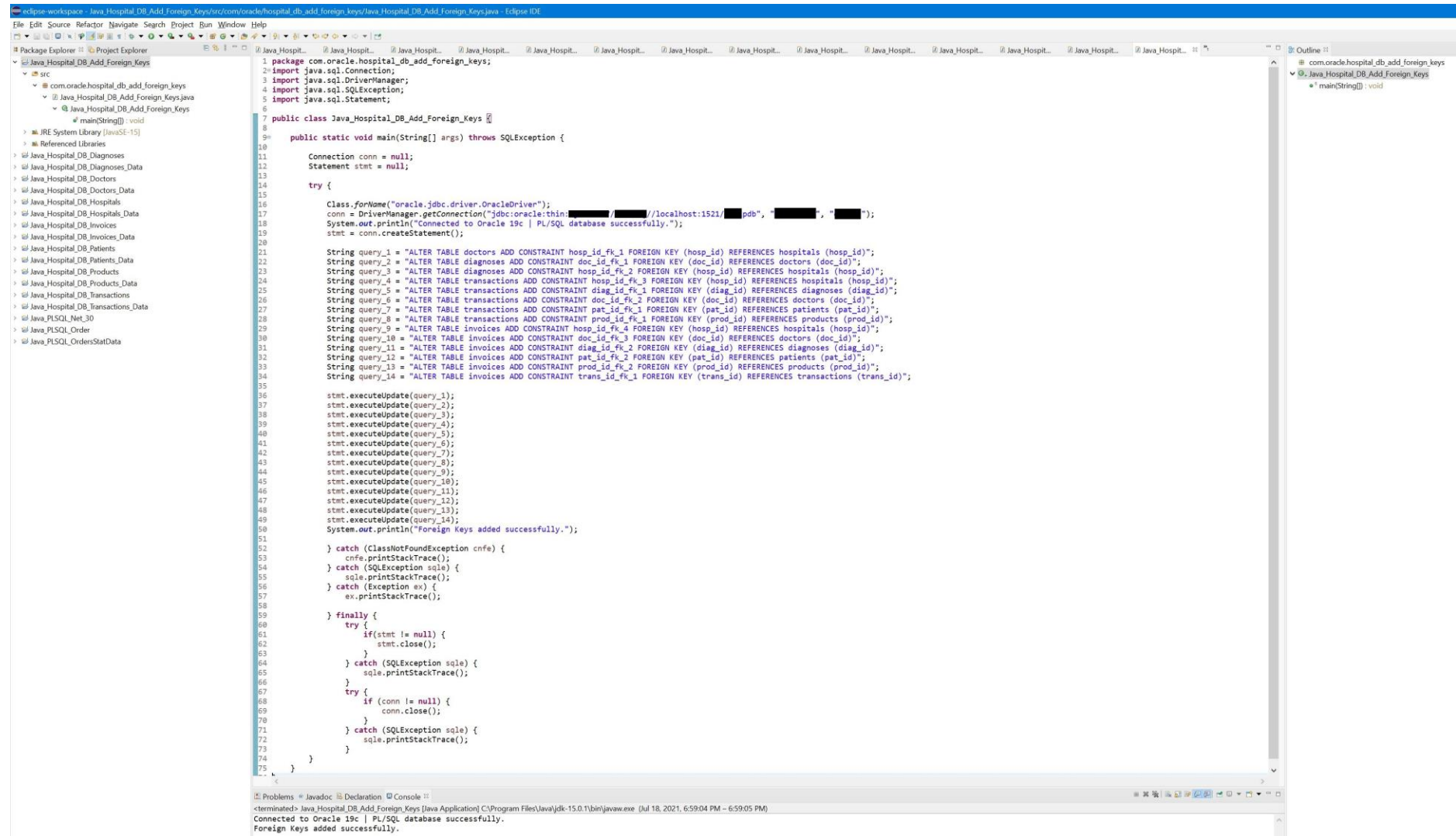
Problems Javadoc Declaration Console  
<terminated> Java\_Hospital\_DB\_Update\_Varchar2 [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (Jul 19, 2021, 6:19:51 PM - 6:19:52 PM)  
Connected to Oracle 19c | PL/SQL database successfully.  
PROD\_DESC column altered.



```
1 package com.oracle.hospital_db_select_data;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.SQLException;
5 import java.sql.Statement;
6 import java.util.Date;
7 import java.sql.ResultSet;
8
9 public class Java_Hospital_DB_Select_Data {
10
11     public static void main(String[] args) throws SQLException {
12
13         Connection conn = null;
14         Statement stmt = null;
15         String query = "SELECT * FROM transactions";
16
17         try {
18             Class.forName("oracle.jdbc.driver.OracleDriver");
19             conn = DriverManager.getConnection("jdbc:oracle:thin:////localhost:1521/ plsqlpdb", " ", " ");
20             System.out.println("Connected to Oracle 19c | PL/SQL database successfully.");
21             stmt = conn.createStatement();
22
23             ResultSet rset = stmt.executeQuery(query);
24
25             while(rset.next()) {
26
27                 int transID = rset.getInt("TRANS_ID");
28                 int hospID = rset.getInt("HOSP_ID");
29                 int diagID = rset.getInt("DIAG_ID");
30                 int docID = rset.getInt("DOC_ID");
31                 int patID = rset.getInt("PAT_ID");
32                 int prodID = rset.getInt("PROD_ID");
33                 int prodQTY = rset.getInt("PROD_QTY");
34                 float prodPrice = rset.getFloat("PROD_PRICE");
35                 float transAmt = rset.getFloat("TRANS_AMT");
36                 String transType = rset.getString("TRANS_TYPE");
37                 Date date = rset.getDate("TRANS_DATE");
38
39                 System.out.print("TRANS_ID: " + transID);
40                 System.out.print(", HOSP_ID: " + hospID);
41                 System.out.print(", DIAG_ID: " + diagID);
42                 System.out.print(", DOC_ID: " + docID);
43                 System.out.print(", PAT_ID: " + patID);
44                 System.out.print(", PROD_ID: " + prodID);
45                 System.out.print(", PROD_QTY: " + prodQTY);
46                 System.out.print(", PROD_PRICE: " + prodPrice);
47                 System.out.print(", TRANS_AMT: " + transAmt);
48                 System.out.print(", TRANS_TYPE: " + transType);
49                 System.out.print(", TRANS_DATE: " + date);
50
51             }
52
53             } catch (ClassNotFoundException cnfe) {
54                 cnfe.printStackTrace();
55             } catch (SQLException sqle) {
56                 sqle.printStackTrace();
57             } catch (Exception ex) {
58                 ex.printStackTrace();
59             } finally {
60                 try {
61                     if(stmt != null) {
62                         stmt.close();
63                     }
64                 } catch (SQLException sqle) {
65                     sqle.printStackTrace();
66                 }
67                 try {
68                     if (conn != null) {
69                         conn.close();
70                     }
71                 }
72             }
73         }
74     }
75 }
```

Connected to Oracle 19c | PL/SQL database successfully.

TRANS_ID	HOSP_ID	DIAG_ID	DOC_ID	PAT_ID	PROD_ID	PROD_QTY	PROD_PRICE	TRANS_AMT	TRANS_TYPE	TRANS_DATE
60006789	10001234	40004567	30003456	20002345	50005678	2	16.0	32.0	Credit Card	2021-07-18
60006790	10001235	40004568	30003457	20002346	50005679	2	35.0	70.0	Credit Card	2021-07-18
60006791	10001236	40004569	30003458	20002347	50005680	2	30.0	60.0	Credit Card	2021-07-18
60006792	10001237	40004570	30003459	20002348	50005681	2	45.0	90.0	Credit Card	2021-07-18
60006793	10001238	40004571	30003460	20002349	50005682	2	17.0	34.0	Credit Card	2021-07-18
60006794	10001239	40004572	30003461	20002350	50005683	3	35.0	105.0	Credit Card	2021-07-18
60006795	10001236	40004569	30003458	20002347	50005680	3	30.0	90.0	Credit Card	2021-07-18
60006797	10001237	40004570	30003459	20002348	50005681	3	45.0	135.0	Credit Card	2021-07-18
60006798	10001238	40004571	30003460	20002349	50005682	3	17.0	51.0	Credit Card	2021-07-18
60006794	10001234	40004567	30003456	20002345	50005678	3	16.0	48.0	Credit Card	2021-07-18



```
1 package com.oracle.hospital_db_add_foreign_keys;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.SQLException;
5 import java.sql.Statement;
6
7 public class Java_Hospital_DB_Add_Foreign_Keys {
8
9     public static void main(String[] args) throws SQLException {
10
11         Connection conn = null;
12         Statement stmt = null;
13
14         try {
15
16             Class.forName("oracle.jdbc.driver.OracleDriver");
17             conn = DriverManager.getConnection("jdbc:oracle:thin:@//localhost:1521/PLSQL", "scott", "tiger");
18             System.out.println("Connected to Oracle 19c | PL/SQL database successfully.");
19             stmt = conn.createStatement();
20
21             String query_1 = "ALTER TABLE doctors ADD CONSTRAINT hosp_id_fk_1 FOREIGN KEY (hosp_id) REFERENCES hospitals (hosp_id)";
22             String query_2 = "ALTER TABLE diagnoses ADD CONSTRAINT doc_id_fk_1 FOREIGN KEY (doc_id) REFERENCES doctors (doc_id)";
23             String query_3 = "ALTER TABLE transactions ADD CONSTRAINT hosp_id_fk_2 FOREIGN KEY (hosp_id) REFERENCES hospitals (hosp_id)";
24             String query_4 = "ALTER TABLE transactions ADD CONSTRAINT prod_id_fk_1 FOREIGN KEY (prod_id) REFERENCES products (prod_id)";
25             String query_5 = "ALTER TABLE transactions ADD CONSTRAINT diag_id_fk_1 FOREIGN KEY (diag_id) REFERENCES diagnoses (diag_id)";
26             String query_6 = "ALTER TABLE transactions ADD CONSTRAINT doc_id_fk_2 FOREIGN KEY (doc_id) REFERENCES doctors (doc_id)";
27             String query_7 = "ALTER TABLE transactions ADD CONSTRAINT pat_id_fk_1 FOREIGN KEY (pat_id) REFERENCES patients (pat_id)";
28             String query_8 = "ALTER TABLE transactions ADD CONSTRAINT prod_id_fk_2 FOREIGN KEY (prod_id) REFERENCES products (prod_id)";
29             String query_9 = "ALTER TABLE transactions ADD CONSTRAINT hosp_id_fk_3 FOREIGN KEY (hosp_id) REFERENCES hospitals (hosp_id)";
30             String query_10 = "ALTER TABLE invoices ADD CONSTRAINT doc_id_fk_3 FOREIGN KEY (doc_id) REFERENCES doctors (doc_id)";
31             String query_11 = "ALTER TABLE invoices ADD CONSTRAINT diag_id_fk_2 FOREIGN KEY (diag_id) REFERENCES diagnoses (diag_id)";
32             String query_12 = "ALTER TABLE invoices ADD CONSTRAINT pat_id_fk_2 FOREIGN KEY (pat_id) REFERENCES patients (pat_id)";
33             String query_13 = "ALTER TABLE invoices ADD CONSTRAINT prod_id_fk_3 FOREIGN KEY (prod_id) REFERENCES products (prod_id)";
34             String query_14 = "ALTER TABLE invoices ADD CONSTRAINT trans_id_fk_1 FOREIGN KEY (trans_id) REFERENCES transactions (trans_id)";
35
36             stmt.executeUpdate(query_1);
37             stmt.executeUpdate(query_2);
38             stmt.executeUpdate(query_3);
39             stmt.executeUpdate(query_4);
40             stmt.executeUpdate(query_5);
41             stmt.executeUpdate(query_6);
42             stmt.executeUpdate(query_7);
43             stmt.executeUpdate(query_8);
44             stmt.executeUpdate(query_9);
45             stmt.executeUpdate(query_10);
46             stmt.executeUpdate(query_11);
47             stmt.executeUpdate(query_12);
48             stmt.executeUpdate(query_13);
49             stmt.executeUpdate(query_14);
50             System.out.println("Foreign Keys added successfully.");
51
52         } catch (ClassNotFoundException cnfe) {
53             cnfe.printStackTrace();
54         } catch (SQLException sqle) {
55             sqle.printStackTrace();
56         } catch (Exception ex) {
57             ex.printStackTrace();
58         }
59
60         finally {
61             try {
62                 if (stmt != null) {
63                     stmt.close();
64                 }
65             } catch (SQLException sqle) {
66                 sqle.printStackTrace();
67             }
68             try {
69                 if (conn != null) {
70                     conn.close();
71                 }
72             } catch (SQLException sqle) {
73                 sqle.printStackTrace();
74             }
75         }
76     }
77 }
```

Problems | Javadoc | Declaration | Console

<terminated> Java\_Hospital\_DB\_Add\_Foreign\_Keys [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (Jul 18, 2021, 6:59:04 PM - 6:59:05 PM)  
Connected to Oracle 19c | PL/SQL database successfully.  
Foreign Keys added successfully.