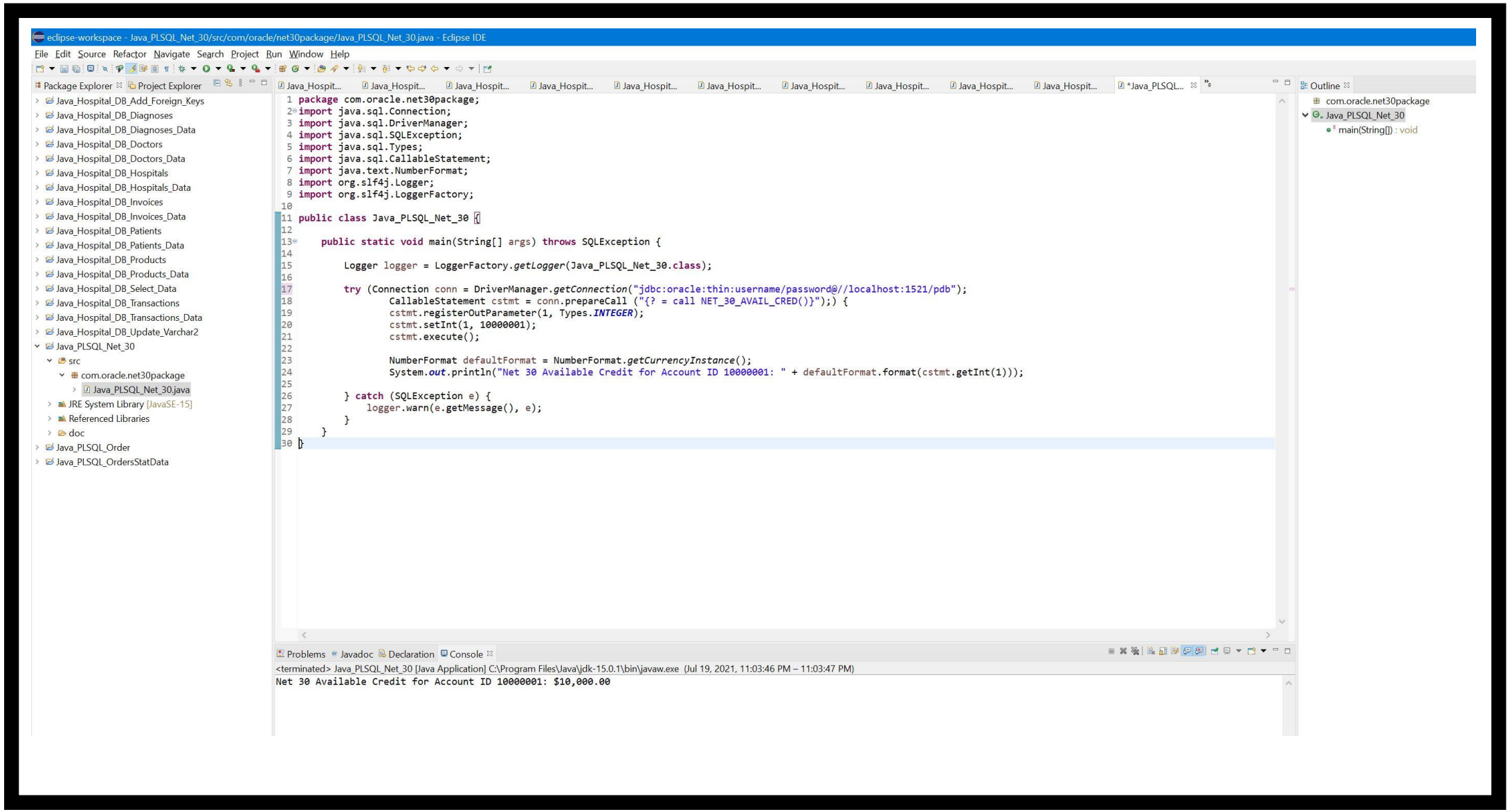
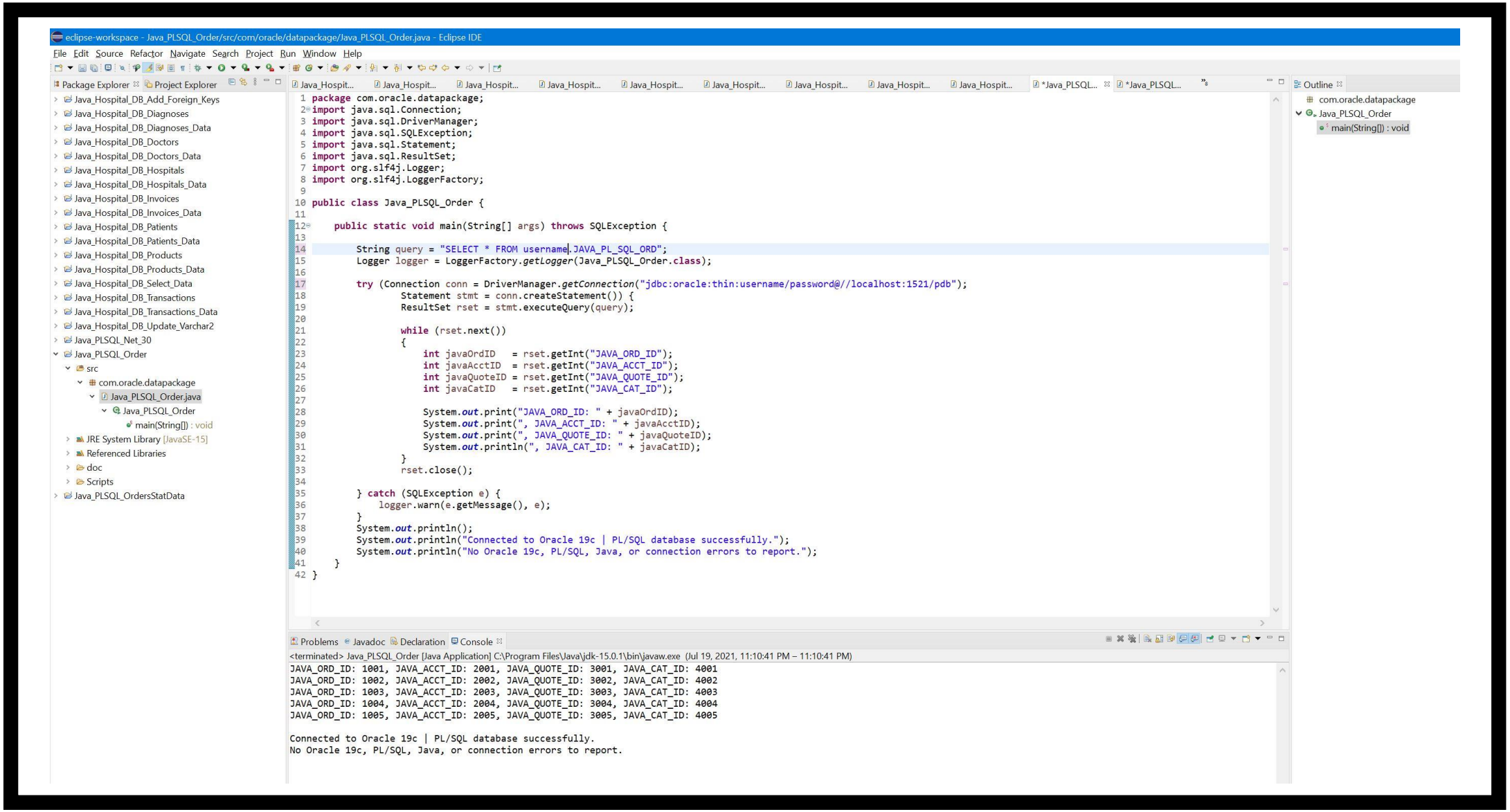
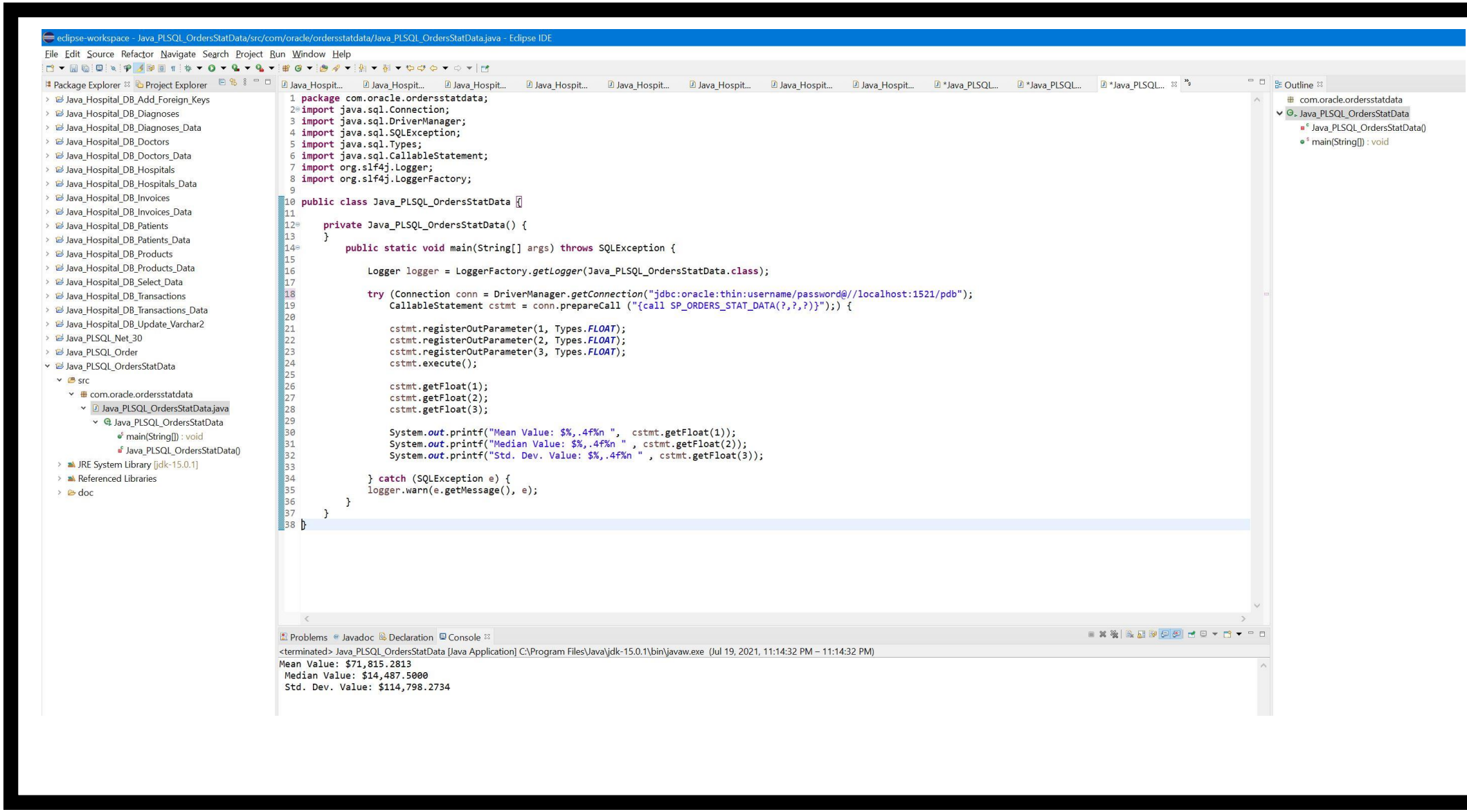
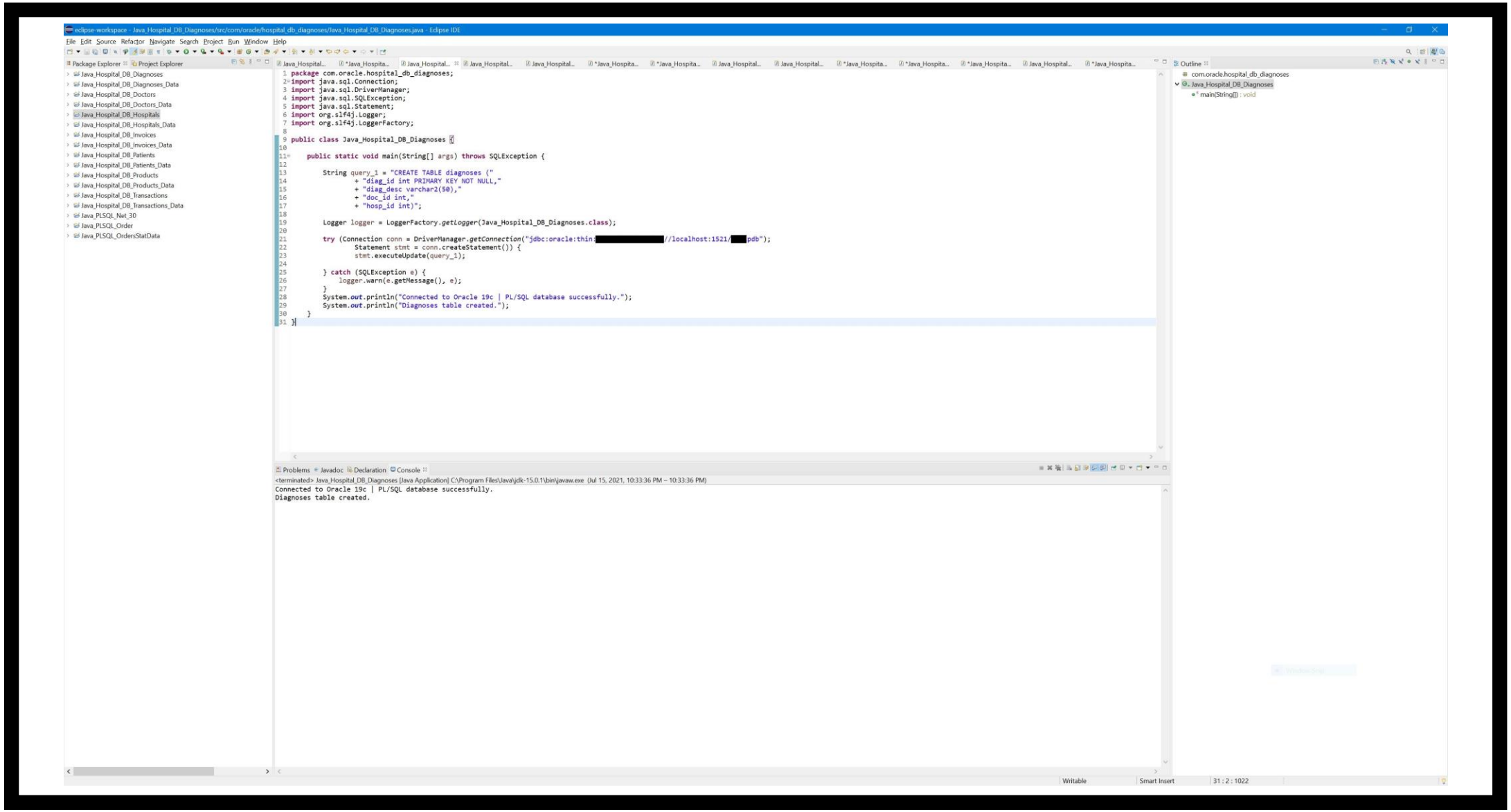
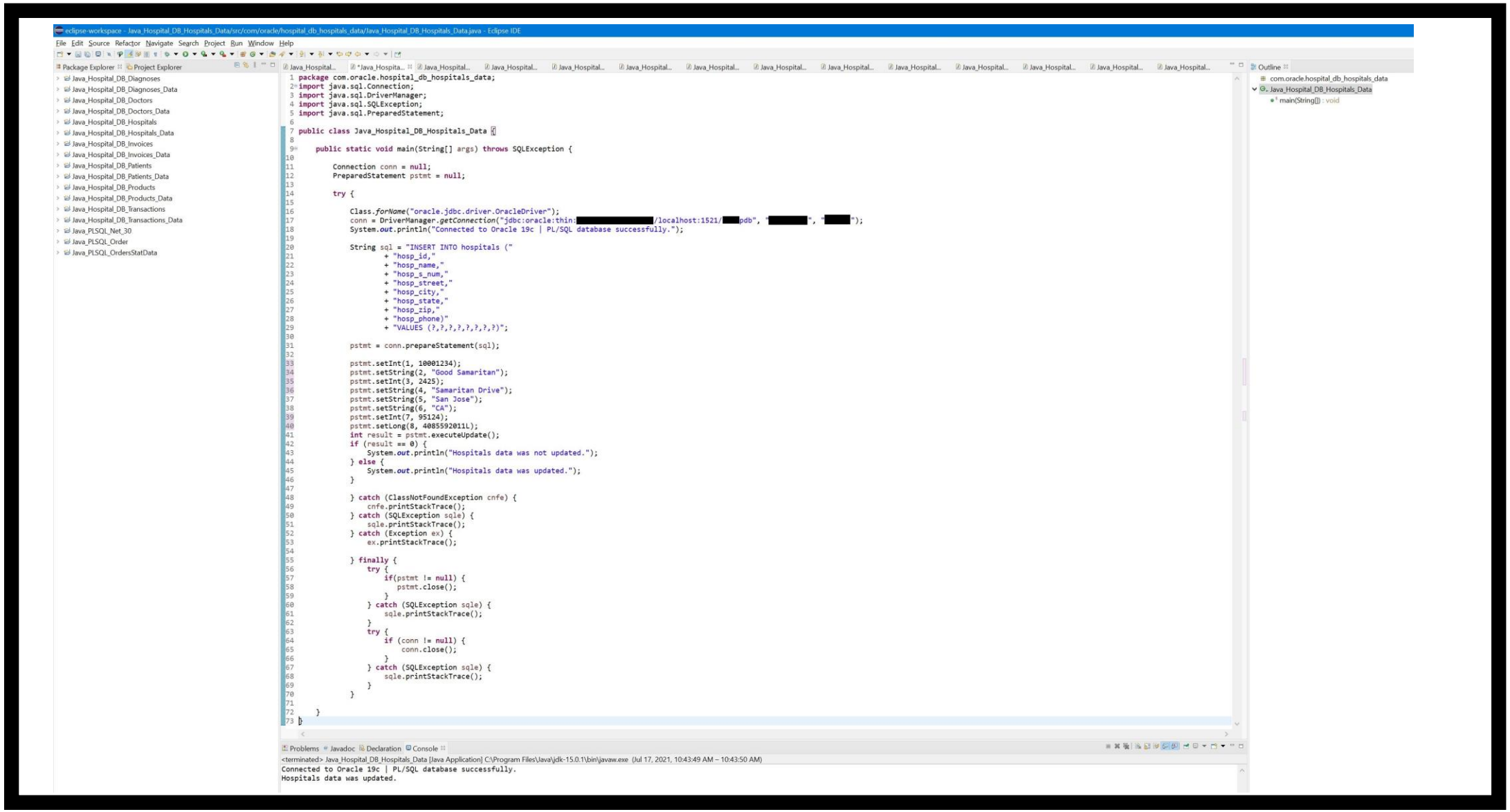

eclipse-workspace - Java_PLSQL_Net_30/src/com/oracle/net30package/Java_PLSQL_Net_30.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
Java_Hospital_DB_Add_Foreign_Keys
Java_Hospital_DB_Diagnoses
Java_Hospital_DB_Diagnoses_Data
Java_Hospital_DB_Doctors
Java_Hospital_DB_Doctors_Data
Java_Hospital_DB_Hospitals
Java_Hospital_DB_Hospitals_Data
Java_Hospital_DB_Invoices
Java_Hospital_DB_Invoices_Data
Java_Hospital_DB_Patients
Java_Hospital_DB_Patients_Data
Java_Hospital_DB_Products
Java_Hospital_DB_Products_Data
Java_Hospital_DB_Select_Data
Java_Hospital_DB_Transactions
Java_Hospital_DB_Transactions_Data
Java_Hospital_DB_Update_Varchar2
Java_PLSQL_Net_30
src
com.oracle.net30package
Java_PLSQL_Net_30.java
JRE System Library [JavaSE-15]
Referenced Libraries
doc
Java_PLSQL_Order
Java_PLSQL_OrdersStatData
*Java_PLSQL...
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 (Connection conn = DriverManager.getConnection("jdbc:oracle:thin:username/password@//localhost:1521/pdb");
18 CallableStatement cstmt = conn.prepareCall ("{? = call NET_30_AVAIL_CRED()}");) {
19 cstmt.registerOutParameter(1, Types.INTEGER);
20 cstmt.setInt(1, 10000001);
21 cstmt.execute();
22
23 NumberFormat defaultFormat = NumberFormat.getCurrencyInstance();
24 System.out.println("Net 30 Available Credit for Account ID 10000001: " + defaultFormat.format(cstmt.getInt(1)));
25
26 } catch (SQLException e) {
27 logger.warn(e.getMessage(), e);
28 }
29 }
30 }
Outline
com.oracle.net30package
Java_PLSQL_Net_30
main(String[]) : void
Problems Javadoc Declaration Console
<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



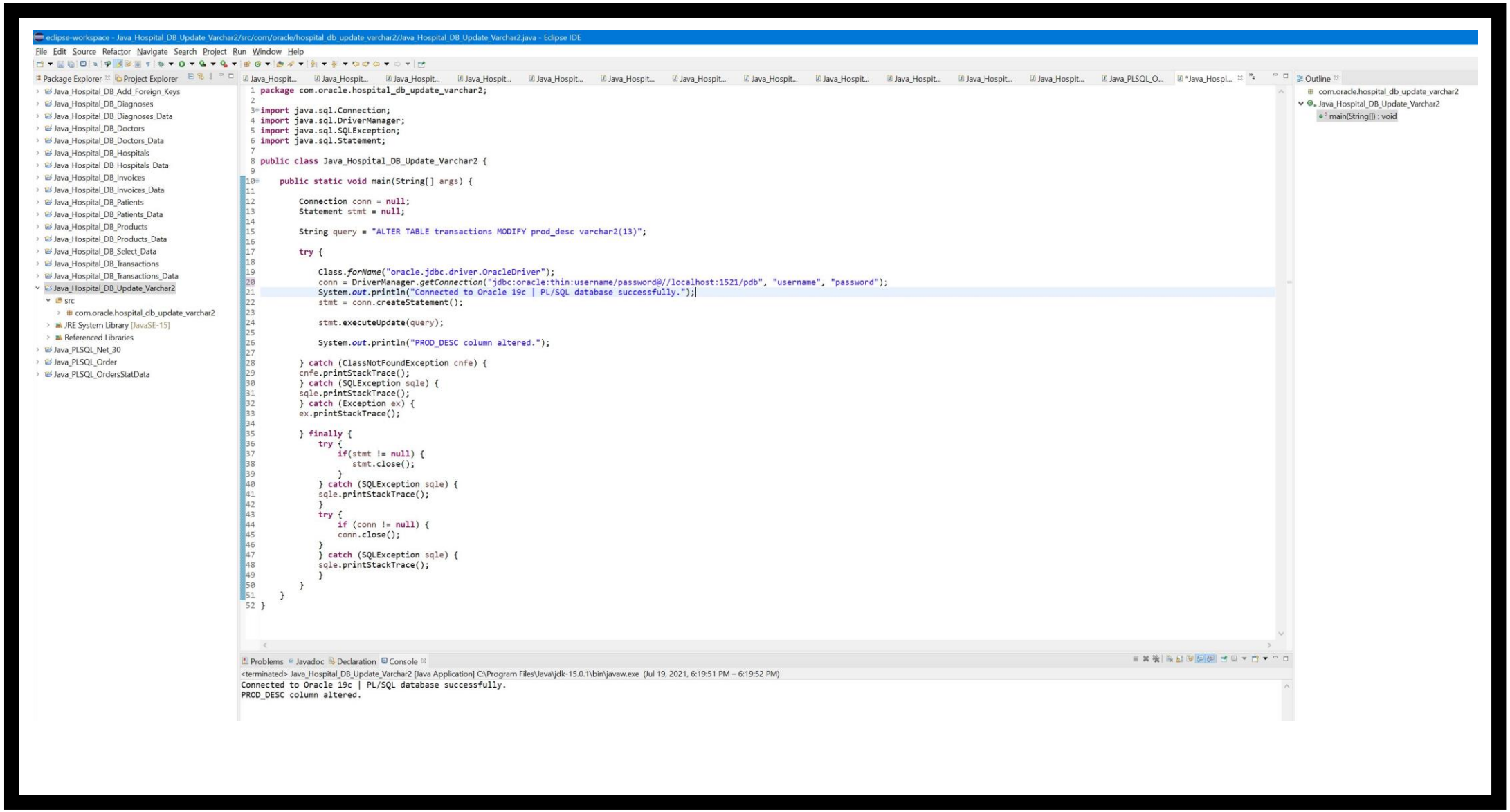

```java
package com.oracle.datapackage;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.ResultSet;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class Java_PLSQL_Order {

    public static void main(String[] args) throws SQLException {

        String query = "SELECT * FROM username.JAVA_PL_SQL_ORD";
        Logger logger = LoggerFactory.getLogger(Java_PLSQL_Order.class);

        try (Connection conn = DriverManager.getConnection("jdbc:oracle:thin:username/password@//localhost:1521/pdb");
                Statement stmt = conn.createStatement()) {
                ResultSet rset = stmt.executeQuery(query);

                while (rset.next())
                {
                    int javaOrdID   = rset.getInt("JAVA_ORD_ID");
                    int javaAcctID  = rset.getInt("JAVA_ACCT_ID");
                    int javaQuoteID = rset.getInt("JAVA_QUOTE_ID");
                    int javaCatID   = rset.getInt("JAVA_CAT_ID");

                    System.out.print("JAVA_ORD_ID: " + javaOrdID);
                    System.out.print(", JAVA_ACCT_ID: " + javaAcctID);
                    System.out.print(", JAVA_QUOTE_ID: " + javaQuoteID);
                    System.out.println(", JAVA_CAT_ID: " + javaCatID);
                }
                rset.close();

        } catch (SQLException e) {
           logger.warn(e.getMessage(), e);
        }
        System.out.println();
        System.out.println("Connected to Oracle 19c | PL/SQL database successfully.");
        System.out.println("No Oracle 19c, PL/SQL, Java, or connection errors to report.");
    }
}
```

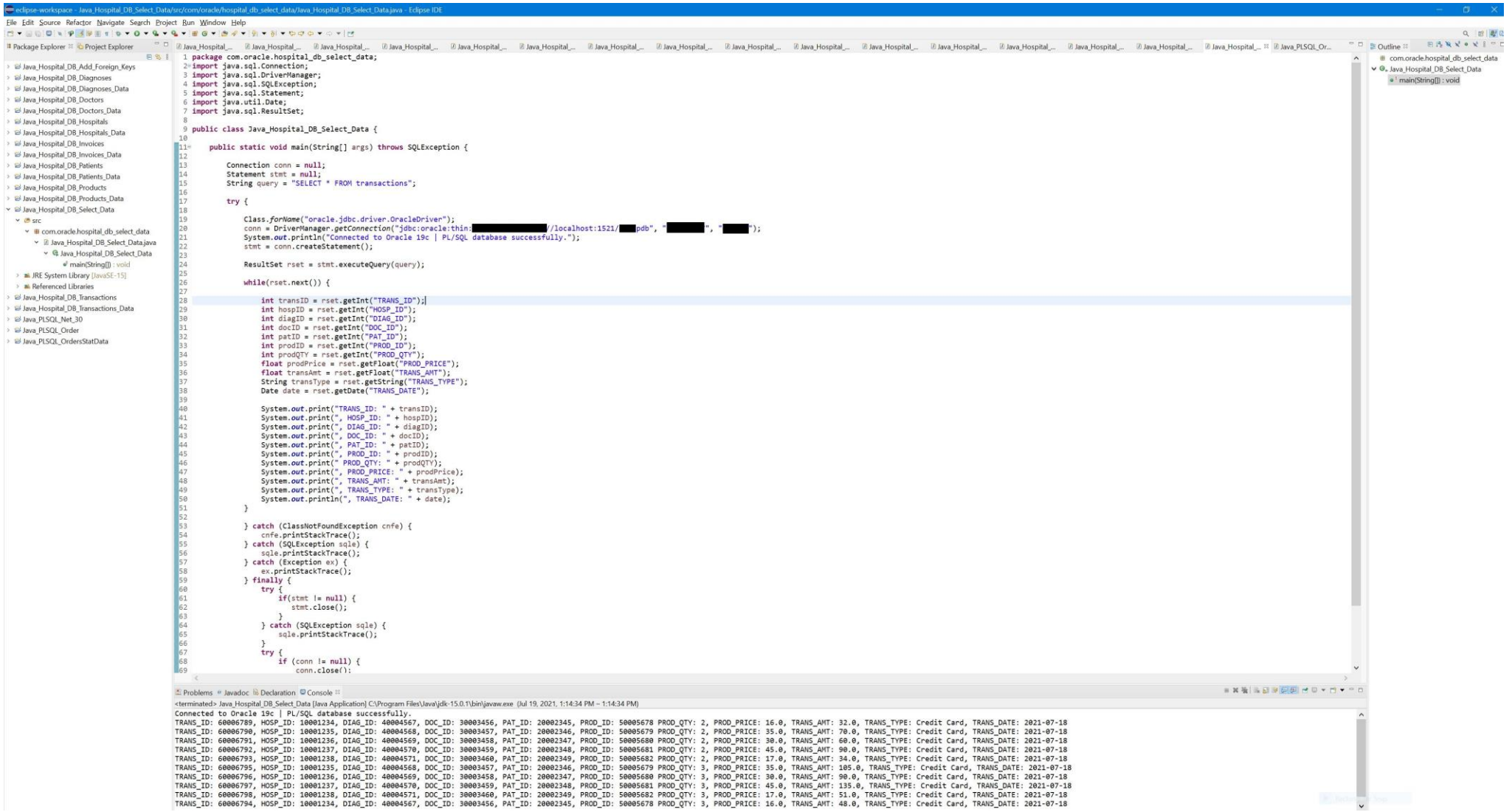

eclipse-workspace - Java_PLSQL_OrdersStatData/src/com/oracle/ordersstatdata/Java_PLSQL_OrdersStatData.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
Java_Hospital_DB_Add_Foreign_Keys
Java_Hospital_DB_Diagnoses
Java_Hospital_DB_Diagnoses_Data
Java_Hospital_DB_Doctors
Java_Hospital_DB_Doctors_Data
Java_Hospital_DB_Hospitals
Java_Hospital_DB_Hospitals_Data
Java_Hospital_DB_Invoices
Java_Hospital_DB_Invoices_Data
Java_Hospital_DB_Patients
Java_Hospital_DB_Patients_Data
Java_Hospital_DB_Products
Java_Hospital_DB_Products_Data
Java_Hospital_DB_Select_Data
Java_Hospital_DB_Transactions
Java_Hospital_DB_Transactions_Data
Java_Hospital_DB_Update_Varchar2
Java_PLSQL_Net_30
Java_PLSQL_Order
Java_PLSQL_OrdersStatData
src
com.oracle.ordersstatdata
Java_PLSQL_OrdersStatData.java
Java_PLSQL_OrdersStatData
main(String[]) : void
Java_PLSQL_OrdersStatData()
JRE System Library [jdk-15.0.1]
Referenced Libraries
doc
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 }
Outline
com.oracle.ordersstatdata
Java_PLSQL_OrdersStatData
Java_PLSQL_OrdersStatData()
main(String[]) : void
Problems Javadoc Declaration Console
<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

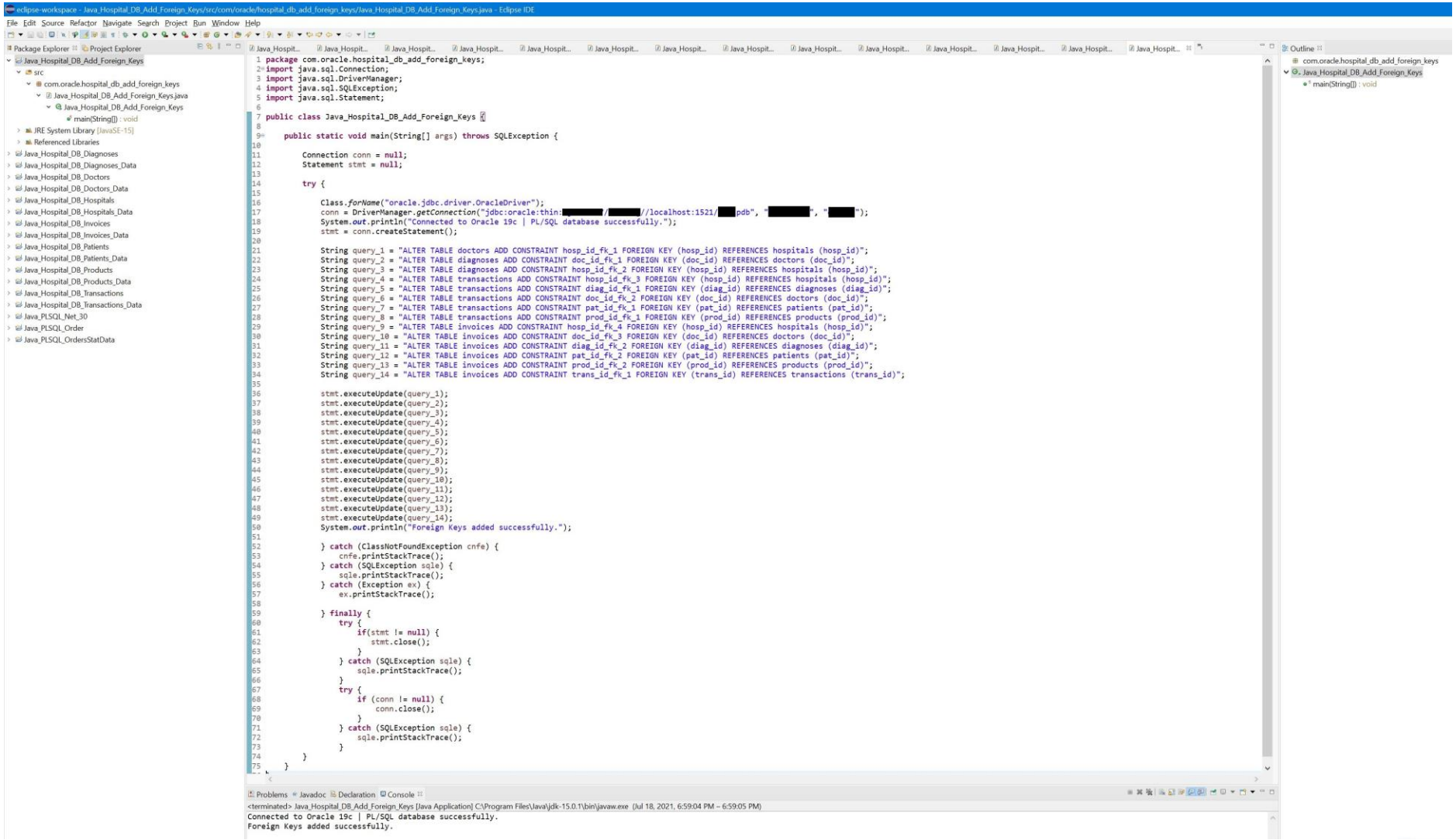

eclipse-workspace - Java_Hospital_DB_Diagnoses/src/com/oracle/hospital_db_diagnoses/Java_Hospital_DB_Diagnoses.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
Java_Hospital_DB_Diagnoses
Java_Hospital_DB_Diagnoses_Data
Java_Hospital_DB_Doctors
Java_Hospital_DB_Doctors_Data
Java_Hospital_DB_Hospitals
Java_Hospital_DB_Hospitals_Data
Java_Hospital_DB_Invoices
Java_Hospital_DB_Invoices_Data
Java_Hospital_DB_Patients
Java_Hospital_DB_Patients_Data
Java_Hospital_DB_Products
Java_Hospital_DB_Products_Data
Java_Hospital_DB_Transactions
Java_Hospital_DB_Transactions_Data
Java_PLSQL_Net_30
Java_PLSQL_Order
Java_PLSQL_OrdersStatData
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:[redacted]//localhost:1521/[redacted]pdb");
22 Statement stmt = conn.createStatement()) {
23 stmt.executeUpdate(query_1);
24
25 } catch (SQLException e) {
26 logger.warn(e.getMessage(), e);
27 }
28 System.out.println("Connected to Oracle 19c | PL/SQL database successfully.");
29 System.out.println("Diagnoses table created.");
30 }
31 }
Problems Javadoc Declaration Console
<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.
Outline
com.oracle.hospital_db_diagnoses
Java_Hospital_DB_Diagnoses
main(String[]) : void
Writable
Smart Insert
31 : 2 : 1022




```
package com.oracle.hospital_db_hospitals_data;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.PreparedStatement;

public class Java_Hospital_DB_Hospitals_Data {

    public static void main(String[] args) throws SQLException {

        Connection conn = null;
        PreparedStatement pstmt = null;

        try {

            Class.forName("oracle.jdbc.driver.OracleDriver");
            conn = DriverManager.getConnection("jdbc:oracle:thin:[redacted]/localhost:1521/[redacted]pdb", "[redacted]", "[redacted]");
            System.out.println("Connected to Oracle 19c | PL/SQL database successfully.");

            String sql = "INSERT INTO hospitals ("
                    + "hosp_id,"
                    + "hosp_name,"
                    + "hosp_s_num,"
                    + "hosp_street,"
                    + "hosp_city,"
                    + "hosp_state,"
                    + "hosp_zip,"
                    + "hosp_phone)"
                    + "VALUES (?,?,?,?,?,?,?,?)";

            pstmt = conn.prepareStatement(sql);

            pstmt.setInt(1, 10001234);
            pstmt.setString(2, "Good Samaritan");
            pstmt.setInt(3, 2425);
            pstmt.setString(4, "Samaritan Drive");
            pstmt.setString(5, "San Jose");
            pstmt.setString(6, "CA");
            pstmt.setInt(7, 95124);
            pstmt.setLong(8, 4085592011L);
            int result = pstmt.executeUpdate();
            if (result == 0) {
                System.out.println("Hospitals data was not updated.");
            } else {
                System.out.println("Hospitals data was updated.");
            }

            } catch (ClassNotFoundException cnfe) {
                cnfe.printStackTrace();
            } catch (SQLException sqle) {
                sqle.printStackTrace();
            } catch (Exception ex) {
                ex.printStackTrace();

            } finally {
                try {
                    if(pstmt != null) {
                        pstmt.close();
                    }
                } catch (SQLException sqle) {
                    sqle.printStackTrace();
                }
                try {
                    if (conn != null) {
                        conn.close();
                    }
                } catch (SQLException sqle) {
                    sqle.printStackTrace();
                }
            }

    }
}
```


eclipse-workspace - Java_Hospital_DB_Update_Varchar2/src/com/oracle/hospital_db_update_varchar2/Java_Hospital_DB_Update_Varchar2.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
Java_Hospital_DB_Add_Foreign_Keys
Java_Hospital_DB_Diagnoses
Java_Hospital_DB_Diagnoses_Data
Java_Hospital_DB_Doctors
Java_Hospital_DB_Doctors_Data
Java_Hospital_DB_Hospitals
Java_Hospital_DB_Hospitals_Data
Java_Hospital_DB_Invoices
Java_Hospital_DB_Invoices_Data
Java_Hospital_DB_Patients
Java_Hospital_DB_Patients_Data
Java_Hospital_DB_Products
Java_Hospital_DB_Products_Data
Java_Hospital_DB_Select_Data
Java_Hospital_DB_Transactions
Java_Hospital_DB_Transactions_Data
Java_Hospital_DB_Update_Varchar2
src
com.oracle.hospital_db_update_varchar2
JRE System Library [JavaSE-15]
Referenced Libraries
Java_PLSQL_Net_30
Java_PLSQL_Order
Java_PLSQL_OrdersStatData
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 } finally {
36 try {
37 if(stmt != null) {
38 stmt.close();
39 }
40 } catch (SQLException sqle) {
41 sqle.printStackTrace();
42 }
43 try {
44 if (conn != null) {
45 conn.close();
46 }
47 } catch (SQLException sqle) {
48 sqle.printStackTrace();
49 }
50 }
51 }
52 }
Outline
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.


eclipse-workspace - Java_Hospital_DB_Select_Data/src/com/oracle/hospital_db_select_data/Java_Hospital_DB_Select_Data.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
Java_Hospital_DB_Add_Foreign_Keys
Java_Hospital_DB_Diagnoses
Java_Hospital_DB_Diagnoses_Data
Java_Hospital_DB_Doctors
Java_Hospital_DB_Doctors_Data
Java_Hospital_DB_Hospitals
Java_Hospital_DB_Hospitals_Data
Java_Hospital_DB_Invoices
Java_Hospital_DB_Invoices_Data
Java_Hospital_DB_Patients
Java_Hospital_DB_Patients_Data
Java_Hospital_DB_Products
Java_Hospital_DB_Products_Data
Java_Hospital_DB_Select_Data
src
com.oracle.hospital_db_select_data
Java_Hospital_DB_Select_Data.java
Java_Hospital_DB_Select_Data
main(String[]) : void
JRE System Library [JavaSE-15]
Referenced Libraries
Java_Hospital_DB_Transactions
Java_Hospital_DB_Transactions_Data
Java_PLSQL_Net_30
Java_PLSQL_Order
Java_PLSQL_OrdersStatData
Outline
com.oracle.hospital_db_select_data
Java_Hospital_DB_Select_Data
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Java_Hospital_DB_Select_Data [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (Jul 19, 2021, 1:14:34 PM – 1:14:34 PM)
Connected to Oracle 19c | PL/SQL database successfully.
TRANS_ID: 60006789, HOSP_ID: 10001234, DIAG_ID: 40004567, DOC_ID: 30003456, PAT_ID: 20002345, PROD_ID: 50005678 PROD_QTY: 2, PROD_PRICE: 16.0, TRANS_AMT: 32.0, TRANS_TYPE: Credit Card, TRANS_DATE: 2021-07-18
TRANS_ID: 60006790, HOSP_ID: 10001235, DIAG_ID: 40004568, DOC_ID: 30003457, PAT_ID: 20002346, PROD_ID: 50005679 PROD_QTY: 2, PROD_PRICE: 35.0, TRANS_AMT: 70.0, TRANS_TYPE: Credit Card, TRANS_DATE: 2021-07-18
TRANS_ID: 60006791, HOSP_ID: 10001236, DIAG_ID: 40004569, DOC_ID: 30003458, PAT_ID: 20002347, PROD_ID: 50005680 PROD_QTY: 2, PROD_PRICE: 30.0, TRANS_AMT: 60.0, TRANS_TYPE: Credit Card, TRANS_DATE: 2021-07-18
TRANS_ID: 60006792, HOSP_ID: 10001237, DIAG_ID: 40004570, DOC_ID: 30003459, PAT_ID: 20002348, PROD_ID: 50005681 PROD_QTY: 2, PROD_PRICE: 45.0, TRANS_AMT: 90.0, TRANS_TYPE: Credit Card, TRANS_DATE: 2021-07-18
TRANS_ID: 60006793, HOSP_ID: 10001238, DIAG_ID: 40004571, DOC_ID: 30003460, PAT_ID: 20002349, PROD_ID: 50005682 PROD_QTY: 2, PROD_PRICE: 17.0, TRANS_AMT: 34.0, TRANS_TYPE: Credit Card, TRANS_DATE: 2021-07-18
TRANS_ID: 60006795, HOSP_ID: 10001235, DIAG_ID: 40004568, DOC_ID: 30003457, PAT_ID: 20002346, PROD_ID: 50005679 PROD_QTY: 3, PROD_PRICE: 35.0, TRANS_AMT: 105.0, TRANS_TYPE: Credit Card, TRANS_DATE: 2021-07-18
TRANS_ID: 60006796, HOSP_ID: 10001236, DIAG_ID: 40004569, DOC_ID: 30003458, PAT_ID: 20002347, PROD_ID: 50005680 PROD_QTY: 3, PROD_PRICE: 30.0, TRANS_AMT: 90.0, TRANS_TYPE: Credit Card, TRANS_DATE: 2021-07-18
TRANS_ID: 60006797, HOSP_ID: 10001237, DIAG_ID: 40004570, DOC_ID: 30003459, PAT_ID: 20002348, PROD_ID: 50005681 PROD_QTY: 3, PROD_PRICE: 45.0, TRANS_AMT: 135.0, TRANS_TYPE: Credit Card, TRANS_DATE: 2021-07-18
TRANS_ID: 60006798, HOSP_ID: 10001238, DIAG_ID: 40004571, DOC_ID: 30003460, PAT_ID: 20002349, PROD_ID: 50005682 PROD_QTY: 3, PROD_PRICE: 17.0, TRANS_AMT: 51.0, TRANS_TYPE: Credit Card, TRANS_DATE: 2021-07-18
TRANS_ID: 60006794, HOSP_ID: 10001234, DIAG_ID: 40004567, DOC_ID: 30003456, PAT_ID: 20002345, PROD_ID: 50005678 PROD_QTY: 3, PROD_PRICE: 16.0, TRANS_AMT: 48.0, TRANS_TYPE: Credit Card, TRANS_DATE: 2021-07-18




```
package com.oracle.hospital_db_add_foreign_keys;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class Java_Hospital_DB_Add_Foreign_Keys {

    public static void main(String[] args) throws SQLException {

        Connection conn = null;
        Statement stmt = null;

        try {

            Class.forName("oracle.jdbc.driver.OracleDriver");
            conn = DriverManager.getConnection("jdbc:oracle:thin:[redacted]/[redacted]//localhost:1521/[redacted].pdb", "[redacted]", "[redacted]");
            System.out.println("Connected to Oracle 19c | PL/SQL database successfully.");
            stmt = conn.createStatement();

            String query_1 = "ALTER TABLE doctors ADD CONSTRAINT hosp_id_fk_1 FOREIGN KEY (hosp_id) REFERENCES hospitals (hosp_id)";
            String query_2 = "ALTER TABLE diagnoses ADD CONSTRAINT doc_id_fk_1 FOREIGN KEY (doc_id) REFERENCES doctors (doc_id)";
            String query_3 = "ALTER TABLE diagnoses ADD CONSTRAINT hosp_id_fk_2 FOREIGN KEY (hosp_id) REFERENCES hospitals (hosp_id)";
            String query_4 = "ALTER TABLE transactions ADD CONSTRAINT hosp_id_fk_3 FOREIGN KEY (hosp_id) REFERENCES hospitals (hosp_id)";
            String query_5 = "ALTER TABLE transactions ADD CONSTRAINT diag_id_fk_1 FOREIGN KEY (diag_id) REFERENCES diagnoses (diag_id)";
            String query_6 = "ALTER TABLE transactions ADD CONSTRAINT doc_id_fk_2 FOREIGN KEY (doc_id) REFERENCES doctors (doc_id)";
            String query_7 = "ALTER TABLE transactions ADD CONSTRAINT pat_id_fk_1 FOREIGN KEY (pat_id) REFERENCES patients (pat_id)";
            String query_8 = "ALTER TABLE transactions ADD CONSTRAINT prod_id_fk_1 FOREIGN KEY (prod_id) REFERENCES products (prod_id)";
            String query_9 = "ALTER TABLE invoices ADD CONSTRAINT hosp_id_fk_4 FOREIGN KEY (hosp_id) REFERENCES hospitals (hosp_id)";
            String query_10 = "ALTER TABLE invoices ADD CONSTRAINT doc_id_fk_3 FOREIGN KEY (doc_id) REFERENCES doctors (doc_id)";
            String query_11 = "ALTER TABLE invoices ADD CONSTRAINT diag_id_fk_2 FOREIGN KEY (diag_id) REFERENCES diagnoses (diag_id)";
            String query_12 = "ALTER TABLE invoices ADD CONSTRAINT pat_id_fk_2 FOREIGN KEY (pat_id) REFERENCES patients (pat_id)";
            String query_13 = "ALTER TABLE invoices ADD CONSTRAINT prod_id_fk_2 FOREIGN KEY (prod_id) REFERENCES products (prod_id)";
            String query_14 = "ALTER TABLE invoices ADD CONSTRAINT trans_id_fk_1 FOREIGN KEY (trans_id) REFERENCES transactions (trans_id)";

            stmt.executeUpdate(query_1);
            stmt.executeUpdate(query_2);
            stmt.executeUpdate(query_3);
            stmt.executeUpdate(query_4);
            stmt.executeUpdate(query_5);
            stmt.executeUpdate(query_6);
            stmt.executeUpdate(query_7);
            stmt.executeUpdate(query_8);
            stmt.executeUpdate(query_9);
            stmt.executeUpdate(query_10);
            stmt.executeUpdate(query_11);
            stmt.executeUpdate(query_12);
            stmt.executeUpdate(query_13);
            stmt.executeUpdate(query_14);
            System.out.println("Foreign Keys added successfully.");

            } catch (ClassNotFoundException cnfe) {
                cnfe.printStackTrace();
            } catch (SQLException sqle) {
                sqle.printStackTrace();
            } catch (Exception ex) {
                ex.printStackTrace();

            } finally {
                try {
                    if(stmt != null) {
                        stmt.close();
                    }
                } catch (SQLException sqle) {
                    sqle.printStackTrace();
                }
                try {
                    if (conn != null) {
                        conn.close();
                    }
                } catch (SQLException sqle) {
                    sqle.printStackTrace();
                }
    }
}
```