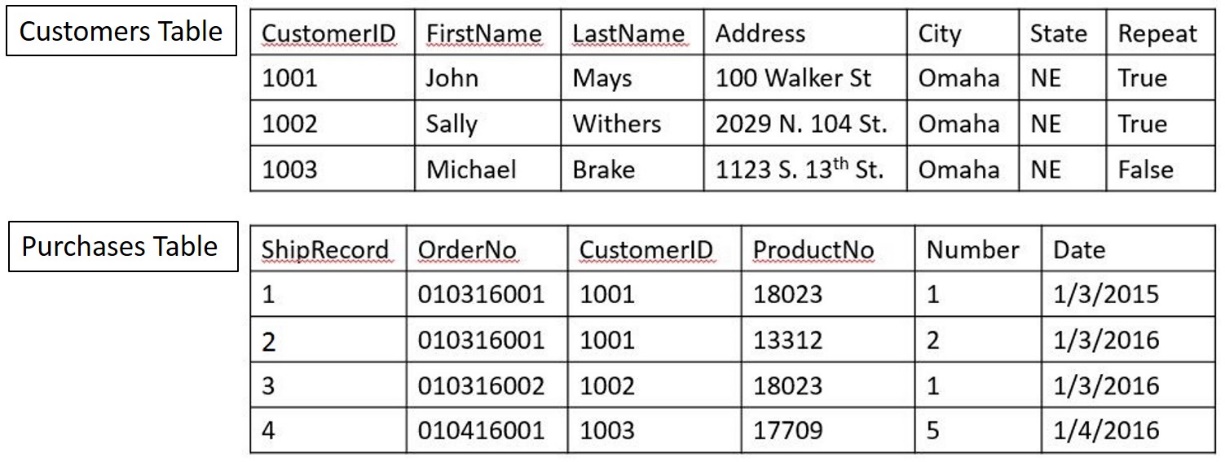
**Java Programming II Final Test Name: \_\_Carter Brehm\_\_\_\_\_\_\_\_\_\_\_**

Database Concepts

|  |
| --- |
| 1. In your own words, describe the purpose of a primary key field.   A primary key field is a unique identifier for the row |
| 1. Describe the purpose of a foreign key field.   It provides a link from a field in one table to a field in another |
| 1. Why are modern databases described as being “relational”?   Because one table does not contain all of the data anymore, there are a variety of types of relationships to split and categorize the data into different tables |
| 1. Modern databases can hold a wide variety of database objects (schemas, tables, queries, reports, views, etc.), but what are the most fundamental objects of a database?   Tables |

The following questions are based on these two tables:



|  |
| --- |
| 1. Do these two tables have a one-to-one, one-to-many, or many-to-many relationship?   One-to-many |
| 1. Circle the primary key field in the Customers table and the foreign key field in the Purchases table. |
| 1. Your database management tool lists the data type of the State column as “Var Char (255)”. Why is that a problem?   It is an invalid datatype (should be varchar, no space) and it performs no validation on the string entered because it has a maximum length of 255, when it could be shortened to 2 characters |
| 1. What would be the best data type for the Repeat column in the Customers table?   boolean |
| 1. What would be the best column for a primary key field in the Purchases table?   CustomerID |

SQL

|  |
| --- |
| 1. Code an SQL statement that will create the Customers table from the previous page. Remember that the table has a primary key and include the statements need to load the indicated records into the table. ( worth double points)   CREATE TABLE Customers  (  CustomerID INTEGER PRIMARY KEY,  FirstName VARCHAR(50),  LastName VARCHAR(50),  Address VARCHAR(50),  City VARCHAR(50),  State VARCHAR(2),  Repeatl BOOL  );  INSERT INTO Customers  VALUES (1001, 'John', 'Mays', '100 Walker St', 'Omaha', 'NE', true);    INSERT INTO Customers  VALUES (1002, 'Sally', 'Withers', '2029 N. 104 St.', 'Omaha', 'NE', true);  INSERT INTO Customers  VALUES (1003, 'Michael', 'Brake', '1123 S. 13th Street', 'Omaha', 'NE', false); |
| 1. Create an SQL statement that returns all of the data in the Customers table.   SELECT \* FROM Customers |
| 1. Create an SQL statement that will return the CustomerID and Address data for all records in the Customers table.   SELECT CustomerID, Address FROM Customers |
| 1. Create an SQL statement that will return the Address, City, and State information for Sally Withers   SELECT \* from Customers WHERE FirstName='Sally' AND LastName='Withers' |
| 1. Create an SQL statement that will change the Sally Withers record in the Customers table so her last name is now Smith.   UPDATE Customers  SET LastName='Smith'  WHERE FirstName='Sally' AND LastName='Withers' |
| 1. Create an SQL statement that will delete the John Mays record from the Customers table.   DELETE FROM Customers  WHERE FirstName='John' AND LastName='Mays' |
| 1. If you delete the John Mays record from Customers, you have a problem in your database. What is it?   The CustomerID is now permanently offset, starting at 1002 |

Java

|  |
| --- |
| 1. Write the code to create a JList object that contains the following data: “Willie Mays”, “Babe Ruth”, “Joe DiMaggio”.   DefaultListModel listModel = new DefaultListModel();  javax.swing.JList list = new javax.swing.JList<>(listModel);  listModel.addElement(“Willie Mays”);  listModel.addElement(“Babe Ruth”);  listModel.addElement(“Joe DiMaggio”); |
| 1. List two kinds of objects you can pass into a JTable constructor that will become the data that displays in the table.   Array or TableModel |
| 1. Write the code to create a JTable object that contains the first names, last names, and birthdays of Presidents James Monroe and James Buchanan. That table should have a three columns and two rows.   String[] columnNames = {"First Name",  "Last Name",  "Birthday"  };  Object[][] data = {  {“James”, “Monroe”, “April 28, 1758”},  {“James”, “Buchanan”, “April 23, 1791”}  };  JTable table = new JTable(data, columnNames); |
| 1. If you want to create a list model class for a JList object, what class should you inherit to do so?   DefaultListModel |
| 1. If you want to create a table model class for JTable object, what class should you inherit to do so?   DefaultTableModel |
| 1. The JOptionPane class has a great many static methods that allow you to create dialog boxes. Name four kinds of dialog boxes that you can create.   Confirm, Input, Message, Option |

The following questions are based on this table



|  |
| --- |
| 1. Write the Java/SQL code to list all of the data in the ClubMembers table. This includes loading the driver (assume you’re using the Derby database) and displaying the data.   Connection dbCon = null;  Statement statement = null;  try {  Class.forName("org.apache.derby.jdbc.EmbeddedDriver");  String myDb = "jdbc:derby://localhost:1527/Tables";  dbCon = DriverManager.getConnection(myDb, "nbuser", "nbuser");  statement = dbCon.createStatement();  } catch (ClassNotFoundException | SQLException exc) {  System.err.println(exc);  System.err.println("Connection failed. Please make sure the database server is running.");  }  ResultSet result = null;  try {  result = statement.executeQuery("SELECT \* from Contractors");  System.out.print(listAll(result));  } catch (SQLException ex) {  System.err.println(ex);  }  StringBuilder sb = new StringBuilder();  try {  ResultSetMetaData metadata = result.getMetaData();  for (int i = 1; i <= metadata.getColumnCount(); i++) {  sb.append(pad(metadata.getColumnName(i), 28));  }  sb.append("\n");  while (result.next()) {  for (int i = 1; i <= metadata.getColumnCount(); i++) {  sb.append(pad(result.getString(i), 25));  sb.append(" | ");  }  sb.append("\n");  }  } catch (SQLException ex) {  System.err.println(ex);  }  System.out.println(sb.toString()); |
| 1. Write the code for a PreparedStatement that will get all of the data from a single column of the ClubMembers table. The column name should be a placeholder   try {  PreparedStatement ps = dbCon.prepareStatement("SELECT ? FROM ClubMembers;");  } catch (SQLException ex) {  System.err.println(ex);  } |
| 1. Write the code for a PreparedStatement that will add a complete record to the ClubMembers table. It should add the values “111”, “Stan Lee”, “07/04/2000”.   try {  PreparedStatement ps = dbCon.prepareStatement("INSERT INTO ClubMembers (MembershipID, Name, DateJoined) VALUES(?,?,?)");  ps.setInt(1, 111);  ps.setString(2, “Stan Lee”);  ps.setString(3, “07/04/2000”);  ps.execute();  } catch (SQLException ex) {  System.err.println(ex);  } |