**Select Statement**

**Write a program to retrieve all the records present in the Book table and display those records in the specified format using the SELECT select.  
Strictly adhere to the Object-Oriented specifications given in the problem statement. All class names, attribute names and method names should be the same as specified in the problem statement.**

**Create a class named Book with the following private attributes/variables.**

|  |  |
| --- | --- |
| **Data type** | **Variable** |
| **Integer** | **id** |
| **String** | **title** |
| **String** | **category** |
| **String** | **author** |
| **Double** | **price** |

**Include appropriate getters and setters.  
Include default and parameterized constructors in the order public Book(Integer id, String title, String category, String author, Double price)**

**Create a class BookdDAO with the following method.**

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| **ArrayList<Book> listBooks()** | **This method is used to list the all the books in the database.** |

**Create a class DBConnection with following method.**

|  |  |
| --- | --- |
| **Method** | **Description** |
| **public static Connection getConnection()** | **This method is used to connect the java application with oracle database. Here register the JDBC driver for the application, configure the database properties(fetch from oracle.properties) and return the connection object.** |

**Create a class Main with main method. In the method, create instances of the above classes and test the above classes.**

**Use the below format to print the details in table :  
System.out.format("%-5s %-20s %-20s %-10s %s\n","Id","Title","Category","Author","Price");**

**oracle.properties :**

**db.url = jdbc:oracle:thin:@localhost:1521:xe  
db.username = root  
db.password = student**

**Use the below code to retrieve the connection details from oracle.properties to establish connection**

**ResourceBundle rb = ResourceBundle.getBundle("oracle");  
String url = rb.getString("db.url");  
String user = rb.getString("db.username");  
String pass = rb.getString("db.password");**

**Table Properties:**

**create table book(id number(10) not null,title VARCHAR2(45) not null,category VARCHAR2(45) not null,  
author VARCHAR2(45) not null,price binary\_double not null,primary key(id));    
Sample Input and Output:**

**List of Books  
Id       Title          Category       Author      Price  
1        Vampire Dairy  Fiction       Chetan      150.0  
2        Harry potter   Witchcraft    Rowling     450.0**

**Select Statement**

**Write a program to retrieve all the records present in the Training table and display those records in the specified format using the SELECT select.  
  
Strictly adhere to the Object-Oriented specifications given in the problem statement. All class names, attribute names and method names should be the same as specified in the problem statement.  
  
Consider a Training class with the following private variables/attributes**

|  |  |
| --- | --- |
| **Data Type** | **Data Member** |
| **String** | **trainingName** |
| **String** | **startDate** |
| **String** | **endDate** |
| **String** | **trainerName** |
| **String** | **venue** |

**The methods for getters, setters, and constructors are given in the template code.**

**Consider TrainingDAO class with the below methods to handle all database related operations**

|  |  |
| --- | --- |
| **Method Name** | **Method Description** |
| **public ArrayList<Training> listAllTraining()** | **This method is used to list and display all training in the database sorted in the ascending order of name** |

**Consider a class DBConnection with the following method.**

|  |  |
| --- | --- |
| **Method** | **Description** |
| **public static Connection getConnection()** | **This method is used to connect the java application with the oracle database. Here register the JDBC driver for the application, configure the database properties(fetch from oracle.properties) and return the connection object.** |

**Consider a class Main with the main method. In the method, create instances of the above classes and test the above classes.**

**Use the below format to print the details in the table :  
System.out.format("%-10s %-10s %-10s %-10s %s\n","Name","Start Date","End Date","Trainer","Venue");  
The Training list should be sorted in the ascending order of Training Name.**

**oracle.properties :**

**db.url = jdbc:oracle:thin:@localhost:1521:xe  
db.username = root  
db.password = student**

**Use the following code snippet to establish DBConnection:**

**ResourceBundle rb = ResourceBundle.getBundle("oracle");  
String url = rb.getString("db.url");  
String user = rb.getString("db.username");  
String pass = rb.getString("db.password");  
 Table Properties:**

**create table training(training\_name varchar2(255) not null,start\_date varchar2(255) not null,  
end\_date varchar2(255) not null,trainer\_name varchar2(255) not null,venue varchar2(255) not null);**

**Sample Input and Output:  
List of Trainings  
Name       Start Date    End Date   Trainer   Venue  
C         28-10-2020    15-12-2020  John      Training hall2  
Java     25-10-2020    31-12-2020  Kelly     Training hall1**

**Search by  Aircraft Manufacturer**

**Neeraja has booked the airplane and she is interested in exploring the things. So she tries to know the models that are provided by the Aircraft manufacturer.  
  
Given the manufacturer name, list the models provided by them.**

**[Note:  Strictly adhere to the object-oriented specifications given as a part of the problem statement. Follow the naming conventions as mentioned. Create separate classes in separate files.]**

**Create  a class AircraftManufacturer with the following attributes.**

|  |  |
| --- | --- |
| **Data Type** | **Variable name** |
| **Integer** | **aircraftManufacturerId** |
| **String** | **name** |

**Include appropirate getters, setters, default and parameterized constructors for the above class  
  
Create  a class Aircraft with the following attributes.**

|  |  |
| --- | --- |
| **Data Type** | **Variable name** |
| **Integer** | **aircraftId** |
| **AircraftManufacturer** | **aircraftManufacturerInstance** |
| **String** | **model** |

**Include appropirate getters, setters, default and parameterized constructors for the above class  
  
Create a class AircraftDAO with the following method.**

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| **ArrayList<String> listModels(String name)** | **This method is used to list the models manufactured by the given manufacturer.** |

**Create a class DBConnection with following method.**

|  |  |
| --- | --- |
| **Method** | **Description** |
| **public static Connection getConnection()** | **This method is used to connect the java application with oracle database.**  **Here register the JDBC driver for the application, configure the database**  **properties(fetch from oracle.properties) and return the connection object.** |

**Create a class Main with main method and call the method  listModels(String name) using  AircraftDAO instance and display the list as shown.**

**oracle.properties :**

**db.url = jdbc:oracle:thin:@localhost:1521:xe  
db.username = root  
db.password = student**

**Use the below code to retrieve the connection details from oracle.properties to establish connection**

**ResourceBundle rb = ResourceBundle.getBundle("oracle");  
String url = rb.getString("db.url");  
String user = rb.getString("db.username");  
String pass = rb.getString("db.password");**

**Table Properties:**

**create table aircraft\_manufacturer(aircraft\_manufacturer\_id number(10) not null,name VARCHAR2(45) not null,primary key(aircraft\_manufacturer\_id));    
create table aircraft(  
aircraft\_id number(10) not null,aircraft\_manufacturer\_id number(10) not null,model VARCHAR2(45) not null,primary key(aircraft\_id),foreign key(aircraft\_manufacturer\_id)  
references aircraft\_manufacturer(aircraft\_manufacturer\_id));**

**Sample Input and Output 1:  
Enter the Aircraft manufacturer :  
Airbus  
The models provided by Airbus are :  
Toulouse A320  
Hamburg A319  
Seville A400M  
  
Sample Input and Output 2:  
Enter the Aircraft manufacturer :  
Boeing  
The models provided by Boeing are :  
Boeing 777  
Boeing 767**

**Update details of Travel Classes**

**Write a java program to update the details of Travel class available in the database and display the list of travel class details in the descending order of names using JDBC drivers.  
  
[Note:  Strictly adhere to the object-oriented specifications given as a part of the problem statement. Follow the naming conventions as mentioned. Create separate classes in separate files.]  
  
Create a class TravelClass with the following attributes.**

|  |  |
| --- | --- |
| **Data Type** | **Variable Name** |
| **String** | **name** |
| **String** | **description** |

**Include appropirate getters, setters, default and parameterized constructors for the above class  
  
Create a class TravelClassDAO with a following method**

|  |  |
| --- | --- |
| **Method name** | **Description** |
| **ArrayList<TravelClass> listAllTravelClassess()** | **This method retrieves the list of travel classes available in the database in the descending order of the travel class name and returns the same.** |
| **void updateDetail(String name, String description)** | **This method update the given description into the database for the given travel class name.** |

**Create a class DBConnection with following method.**

|  |  |
| --- | --- |
| **Method** | **Description** |
| **public static Connection getConnection()** | **This method is used to connect the java application with oracle database. Here register the JDBC driver for the application, configure the database properties(fetch from oracle.properties) and return the connection object.** |

**Create a class Main with main method and call the methods of TravelClassDAO and display the list as shown in the main method.  
  
Use the below format to print the details in table :  
System.out.format("%-25s %s\n","Name","Description");**

**oracle.properties :**

**db.url = jdbc:oracle:thin:@localhost:1521:xe  
db.username = root  
db.password = student**

**Use the below code to retrieve the connection details from oracle.properties to establish connection**

**ResourceBundle rb = ResourceBundle.getBundle("oracle");  
String url = rb.getString("db.url");  
String user = rb.getString("db.username");  
String pass = rb.getString("db.password");**

**Table Properties:**

**create table travel\_class(  
id number(10) not null,  
name VARCHAR2(45) not null,  
description CLOB not null,  
primary key(id)  
);**

**Sample Input and Output:  
Enter the name of TravelClass :  
Economy Class  
Enter the description to update :  
Lowest travel class of seating in flight travel.  
Updated List of Travel Classes  
Name                      Description                     
Premium Economy Class     Positioning in price, comfort, and amenities, this travel class is leveled between economy class and business class.  
Economy Class             Lowest travel class of seating in flight travel.  
Business Class            Intermediate level of service between economy class and first class.**

**User-Delete**

**The users who enter into 20 Ideas for Vision 2020 must be active and must keep posting their ideas.If the administrator finds out that the user is not active the admin can delete the user record from the table. Help the admin to delete the user record from the table.  
[Note:  Strictly adhere to the object-oriented specifications given as a part of the problem statement. Follow the naming conventions as mentioned. Create separate classes in separate files.]  
Create a User with following private attributes.**

|  |  |
| --- | --- |
| **Data Type** | **Variable** |
| **Integer** | **id** |
| **String** | **name** |
| **String** | **email** |
| **String** | **password** |
| **Integer** | **age** |
| **String** | **role** |
| **Date** | **createdDate** |
| **String** | **status** |

**Include appropriate getters, setters, default and parameterized constructors for the above class.**

**Create a class UserDAO with following methods**

|  |  |
| --- | --- |
| **Method Name** | **Method Description** |
| **ArrayList<User> listUsers()** | **This method returns the list of users available in the database** |
| **void deleteUser(Integer id)** | **This method will delete the user from the database using the given id** |

**Create a class DBConnection with following method**

|  |  |
| --- | --- |
| **Method** | **Description** |
| **public static Connection getConnection()** | **This method is used to connect the java application with oracle database. Here register the JDBC driver for the application,configure the database properties(fetch from oracle.properties) and return the connection object.** |

**Create a class Main with main method and call the methods of UserDAO and display the list as shown.**

**Output Format:  
Use Java Format Specifier to display the user details  
 System.out.format("%-15s %-15s %-15s %-15s %-15s %-15s %-15s %s\n","Id","Name","Email","Password","Age","Role","CreatedDate","Status");  
 Table Properties:**

**create table "user"(id number(10) GENERATED ALWAYS AS IDENTITY not null,  
name varchar2(50) not null,email varchar2(50) not null,password varchar2(50) not null,age number(10) not null,role varchar2(50) not null,created\_date date not null,status varchar2(50) not null,primary key(id));**

**oracle.properties :**

**db.url = jdbc:oracle:thin:@localhost:1521:xe  
db.username = root  
db.password = student**

**Use the below code to retrieve the connection details from oracle.properties to establish connection**

**ResourceBundle rb = ResourceBundle.getBundle("oracle");  
String url = rb.getString("db.url");  
String user = rb.getString("db.username");  
String pass = rb.getString("db.password");**

**Sample Input and Output  
[All text in bold are input and the remaining are output]**

**Before the Delete  
Id              Name            Email           Password        Age             Role            CreatedDate     Status           
1               Asha            ash@a.com       as123           18              user            2015-10-13      Approved         
2               Rahul           rh@a.com        rh@123          15              user            2015-10-14      Approved         
3               Ravi            rv@a.com        rv@98           20              user            2015-10-14      pending          
Enter the Id :  
1  
After the Delete  
Id              Name            Email           Password        Age             Role            CreatedDate     Status           
2               Rahul           rh@a.com        rh@123          15              user            2015-10-14      Approved         
3               Ravi            rv@a.com        rv@98           20              user            2015-10-14      pending**

**Contest Enrollment Form**

Design a page for the contest registration form as shown in the screenshot.  
  
**Sample Screenshot 1 :**  
  
Graphical user interface, text, application

Description automatically generated  
  
**Sample Screenshot 2 :**   
  
**A picture containing table

Description automatically generated**  
  
**Content :**  
  
**City Select box option**

1. Select the City
2. Madurai
3. Chennai
4. Coimbatore
5. Bangalore
6. Hyderabad

**State Select box option**

1. Select the State
2. Tamilnadu
3. Andra Pradesh
4. Telangana
5. Karnataka
6. Madhya Pradesh

**Country Select Box option**

1. Select the Country
2. India
3. US
4. Russia
5. UK
6. Spain?

**Languages**

1. C
2. C++
3. Java
4. Python
5. Perl
6. Ruby
7. C#
8. HTML
9. CSS
10. Javascript

**Note :**  
  
Content of the page should be present as shown in the screenshot.  
Kindly refer to the content which is given as a part of the description.