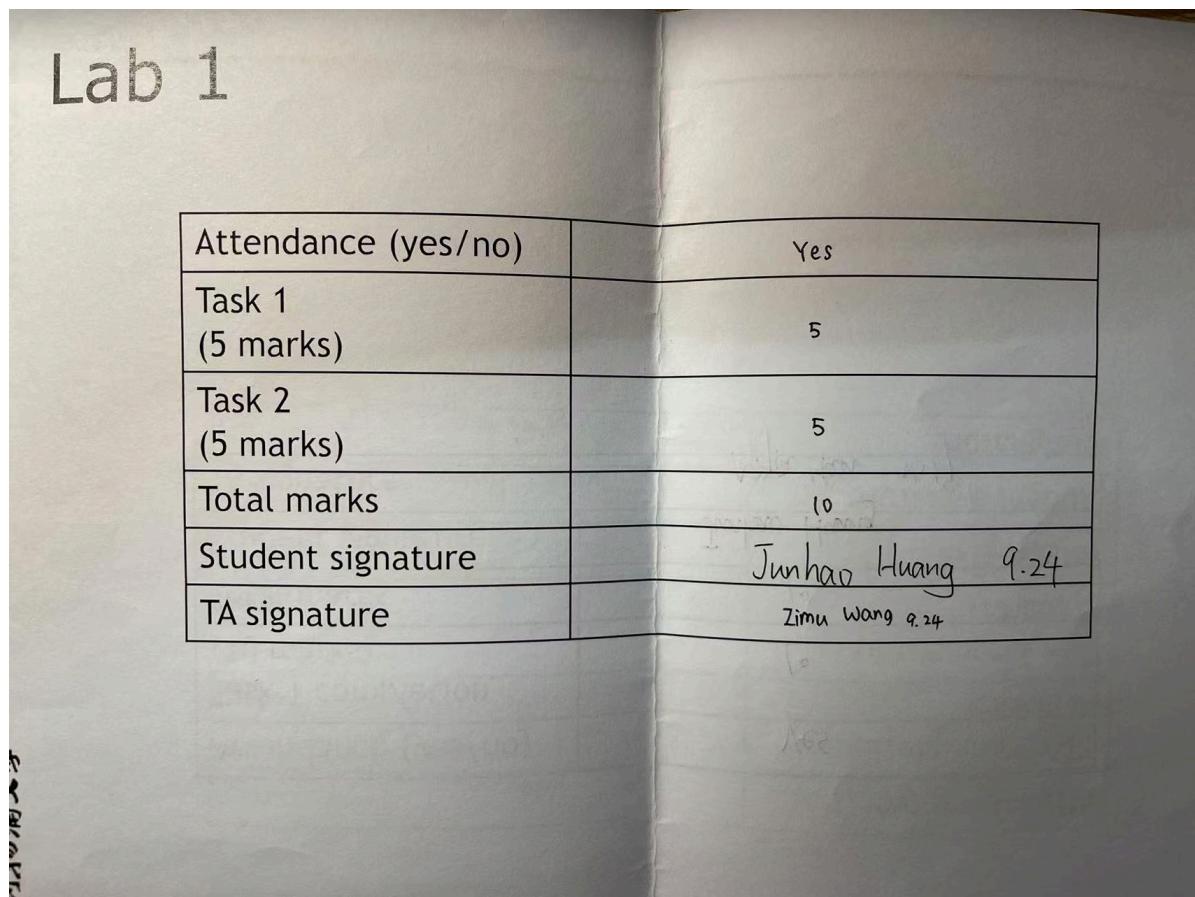


# Lab 1 JDBC



A screenshot of a web browser showing the phpMyAdmin interface. The user is connected to the database 'junhaohuang2202' and is viewing the 'Orf Motif' table. The SQL query executed was:

```
SELECT * FROM Orf_Motif WHERE num=7;
```

The results of the query are displayed in a table:

orf	acc_num	num	pos	len	mmatch
YBL113c	PS00001	7	139	4	NSST
YBL113c	PS00005	7	577	3	S <small>r</small> R
YBL113c	PS00006	7	157	4	T <small>t</small> E
YBL113c	PS00008	7	693	6	GVtpSK
YBL111c	PS00005	7	229	3	T <small>e</small> R
YBL111c	PS00006	7	360	4	S <small>s</small> fE
YBL106c	PS00005	7	484	3	S <small>v</small> R
YBL106c	PS00006	7	276	4	S <small>i</small> F
YBL106c	PS00008	7	600	6	GAvsAI
YBL105c	PS00001	7	979	4	NRTS
YBL105c	PS00005	7	548	3	T <small>k</small> R
YBL105c	PS00006	7	274	4	TvyD
YBL105c	PS00008	7	491	6	G <small>T</small> kwCC
YBL104c	PS00001	7	314	4	NGTS
YBL104c	PS00005	7	466	3	S <small>s</small> K
YBL104c	PS00006	7	350	4	S <small>f</small> fE
YBL104c	PS00008	7	228	6	G <small>s</small> gaAS
YBL101c	PS00001	7	399	4	NNSN
控制台 PS00004					

mysql\_百度搜索 | C mysql安装配置教程-CSDN搜索 | Xian Jiaotong-Liver... | csse-mysql.xjtu.edu.cn/127.0.0.1 | Bilibili

phpMyAdmin

近期访问 表收藏夹

information\_schema  
junhaohuang2202  
新建  
module\_enrollment  
orf\_motif  
students  
teachers

在表 junhaohuang2202.Orf\_Motif 运行 SQL 查询:

```
1 CREATE INDEX numIndex ON Orf_Motif (num);
```

orf acc\_num num pos len mmatch

SELECT \* SELECT INSERT UPDATE DELETE 清除 格式 获取自动保存的查询

将此 SQL 查询加为书签: [输入框]

语句界符: ; 在此再次显示此查询 保留查询框 完成后回滚 启用外键约束 执行

隐藏查询框

MySQL 返回的查询结果为空 (即零行)。 (查询花费 0.0024 秒。)

CREATE INDEX numIndex ON Orf\_Motif (num);

[编辑内嵌] [编辑] [创建 PHP 代码]

控制台

25°C 多云 16:58 2024/9/24

mysql\_百度搜索 | C mysql安装配置教程-CSDN搜索 | Xian Jiaotong-Liver... | csse-mysql.xjtu.edu.cn/127.0.0.1 | Bilibili

phpMyAdmin

近期访问 表收藏夹

information\_schema  
junhaohuang2202  
新建  
module\_enrollment  
orf\_motif  
students  
teachers

在表 junhaohuang2202.Orf\_Motif 运行 SQL 查询:

```
1 SELECT * FROM Orf_Motif  
2 WHERE num=7;
```

orf acc\_num num pos len mmatch

SELECT \* FROM Orf\_Motif WHERE num=7;

性能分析 [编辑内嵌] [编辑] [解析 SQL] [创建 PHP 代码] [刷新]

1 < > >> 行数: 25 过滤行: 在表中搜索 按索引排序: 无

控制台

25°C 多云 17:00 2024/9/24

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project:** DatabaseTest
- File:** JDBCIndex.java
- Code Content:**

```

17  public class JDBCIndex {
18
19      public static void main(String[] args) {
20
21          System.out.println(" first col" + "\t" + " second col");
22          System.out.println("-----");
23
24          //print actual data from the query
25          while (rs.next()) {
26              first = rs.getString( columnLabel );
27              System.out.println(rs.getString( columnLabel ) + "acc_num") + "\t" + first);
28          }
29          System.out.println("End query without an index.");
30          System.out.println("Query without an index takes " + (end - begin) + "milliseconds");
31
32      //END TEST QUERY WITHOUT AN INDEX
33
34      //TEST QUERY WITH AN INDEX
35      //prepare query with an index

```

- Run:** JDBCIndex
- Output:**

```

D:\Development\Java\Java-17\bin\java.exe -javaagent:D:\Development\IntelliJ_IDEA\lib\idea_rt.jar=35212:D:\Development\IntelliJ_IDEA\bin -Dfile.encoding=UTF-8 -classpath C:\Users\YBL113c\IdeaProjects\DatabaseTest\src Lab1 JDBCIndex
Succeeded connecting to the Database.
Started index on num dropped.
Start Query without an index.
-----
the result is:
-----
first col    second col
-----
PS00001 YBL113c
PS00005 YBL113c
PS00006 YBL113c
PS00008 YBL113c
PS00005 YBL111c
PS00006 YBL111c
PS00005 YBL106c
PS00006 YBL106c
PS00008 YBL106c
PS00001 YBL105c
PS00005 YBL105c

```

- System Bar:** 25°C, 多云, 17:17, 2024/9/24

## Lab2 JTA

Handwritten student information card:

Attendance (yes/no)	Yes
Task 1 completion (10 marks)	10
Total marks	10
Student signature	Junhao Huang
TA signature	Mia Han 11/19

Handwritten notes on the back of the card:

- 西文打字机用文字
- 9/22/24

phpMyAdmin

服务器: 127.0.0.1 数据库: junhaohuang2202 表: t\_account1

正在显示第 0 - 1 行 (共 2 行, 查询花费 0.0003 秒。)

```
SELECT * FROM `t_account1`
```

性能分析 [ 编辑内嵌 ] [ 编辑 ] [ 解析 SQL ] [ 创建 PHP 代码 ] [ 刷新 ]

显示全部 行数: 25 过滤行: 在表中搜索

额外选项

id	balance
1	1100
2	2000

显示全部 行数: 25 过滤行: 在表中搜索

查询结果操作 打印 复制到剪贴板 导出 显示图表 新建视图

将此 SQL 查询加为书签 标签:  让所有用户均可访问此书签

将此 SQL 查询加为书签

控制台

18:06 2024/11/19

phpMyAdmin

服务器: 127.0.0.1 数据库: junhaohuang2202 表: t\_account1

正在显示第 0 - 1 行 (共 2 行, 查询花费 0.0157 秒。)

```
SELECT * FROM `t_account1`
```

性能分析 [ 编辑内嵌 ] [ 编辑 ] [ 解析 SQL ] [ 创建 PHP 代码 ] [ 刷新 ]

显示全部 行数: 25 过滤行: 在表中搜索

额外选项

id	balance
1	1200
2	2000

显示全部 行数: 25 过滤行: 在表中搜索

查询结果操作 打印 复制到剪贴板 导出 显示图表 新建视图

将此 SQL 查询加为书签 标签:  让所有用户均可访问此书签

将此 SQL 查询加为书签

控制台

18:15 2024/11/19

phpMyAdmin

服务器: 127.0.0.1 > 数据库: junhaohuang2202 > 表: t\_account2

正在显示第 0 - 1 行 (共 2 行, 查询花费 0.0002 秒.)

SELECT \* FROM `t\_account2`

性能分析 [编辑内嵌] [编辑] [解析 SQL] [创建 PHP 代码] [刷新]

显示全部 行数: 25 过滤行: 在表中搜索

额外选项

id	balance
1	600
2	2000

显示全部 行数: 25 过滤行: 在表中搜索

查询结果操作 打印 复制到剪贴板 导出 显示图表 新建视图

将此 SQL 查询加为书签 标签:  让所有用户均可访问此书签

将此 SQL 查询加为书签

控制台

18:06 2024/11/19

The screenshot shows the phpMyAdmin interface for a MySQL database named 'junhaohuang2202'. The current table is 't\_account2'. A single query is run: 'SELECT \* FROM `t\_account2`'. The results show two rows: one with id 1 and balance 600, and another with id 2 and balance 2000. The interface includes standard navigation and search tools, as well as options to print or export the data.

phpMyAdmin

服务器: 127.0.0.1 > 数据库: junhaohuang2202 > 表: t\_account2

正在显示第 0 - 1 行 (共 2 行, 查询花费 0.0097 秒.)

SELECT \* FROM `t\_account2`

性能分析 [编辑内嵌] [编辑] [解析 SQL] [创建 PHP 代码] [刷新]

显示全部 行数: 25 过滤行: 在表中搜索

额外选项

id	balance
1	500
2	2000

显示全部 行数: 25 过滤行: 在表中搜索

查询结果操作 打印 复制到剪贴板 导出 显示图表 新建视图

将此 SQL 查询加为书签 标签:  让所有用户均可访问此书签

将此 SQL 查询加为书签

控制台

18:17 2024/11/19

This screenshot is nearly identical to the one above, showing the same database and table. However, the data in the table has been modified: the row with id 1 now has a balance of 500, while the row with id 2 remains at 2000. The rest of the interface and status bar are also identical.

C:\Windows\System32\cmd.exe + ^

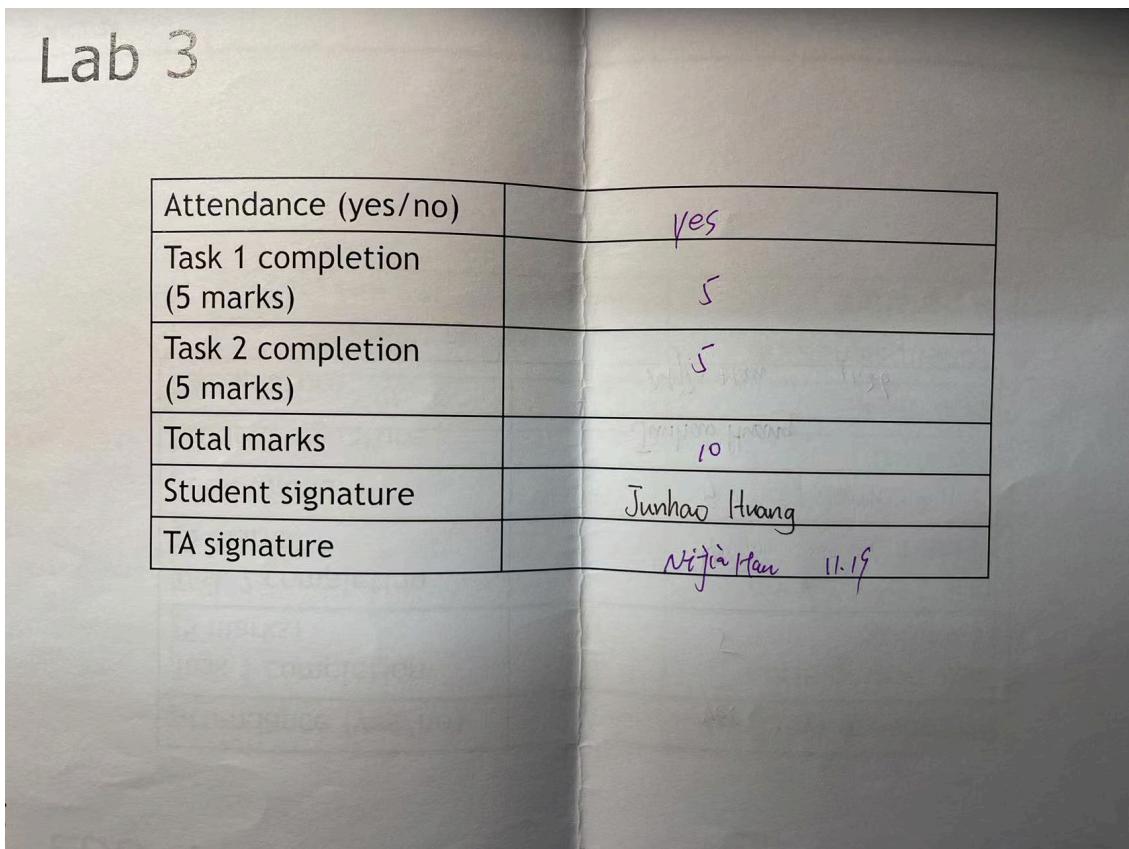
Microsoft Windows [版本 10.0\_22631.4460]  
(c) Microsoft Corporation. 保留所有权利。

D:\Projects\Java\_Projects\src\Lab2>java -cp mysql-connector-j-8.0.33.jar; FundTransferJTA.java  
OK!

D:\Projects\Java\_Projects\src\Lab2>

The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The user has navigated to the directory 'D:\Projects\Java\_Projects\src\Lab2'. They have run the command 'java -cp mysql-connector-j-8.0.33.jar; FundTransferJTA.java', which resulted in the output 'OK!'. The prompt then changes to 'D:\Projects\Java\_Projects\src\Lab2>'.

# Lab 3 XML Processing



## DOM

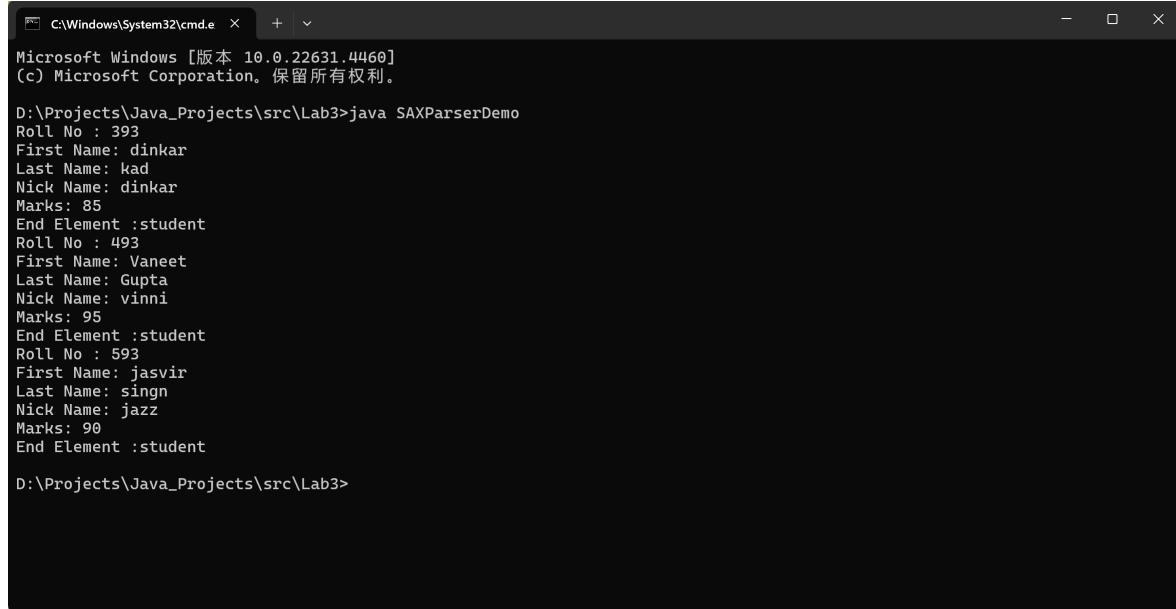
```
C:\Windows\System32\cmd.e + 
Nick Name: jazz
Marks: 90
End Element :student
D:\Projects\Java_Projects\src\Lab3>java DOMParserDemo
Root element :class
-----
Current Element:student
Student roll no: 393
First Name: dinkar
Last Name: kad
Nick Name: dinkar
Marks: 85

Current Element:student
Student roll no: 493
First Name: Vaneet
Last Name: Gupta
Nick Name: vinni
Marks: 95

Current Element:student
Student roll no: 593
First Name: jasvir
Last Name: singn
Nick Name: jazz
Marks: 90

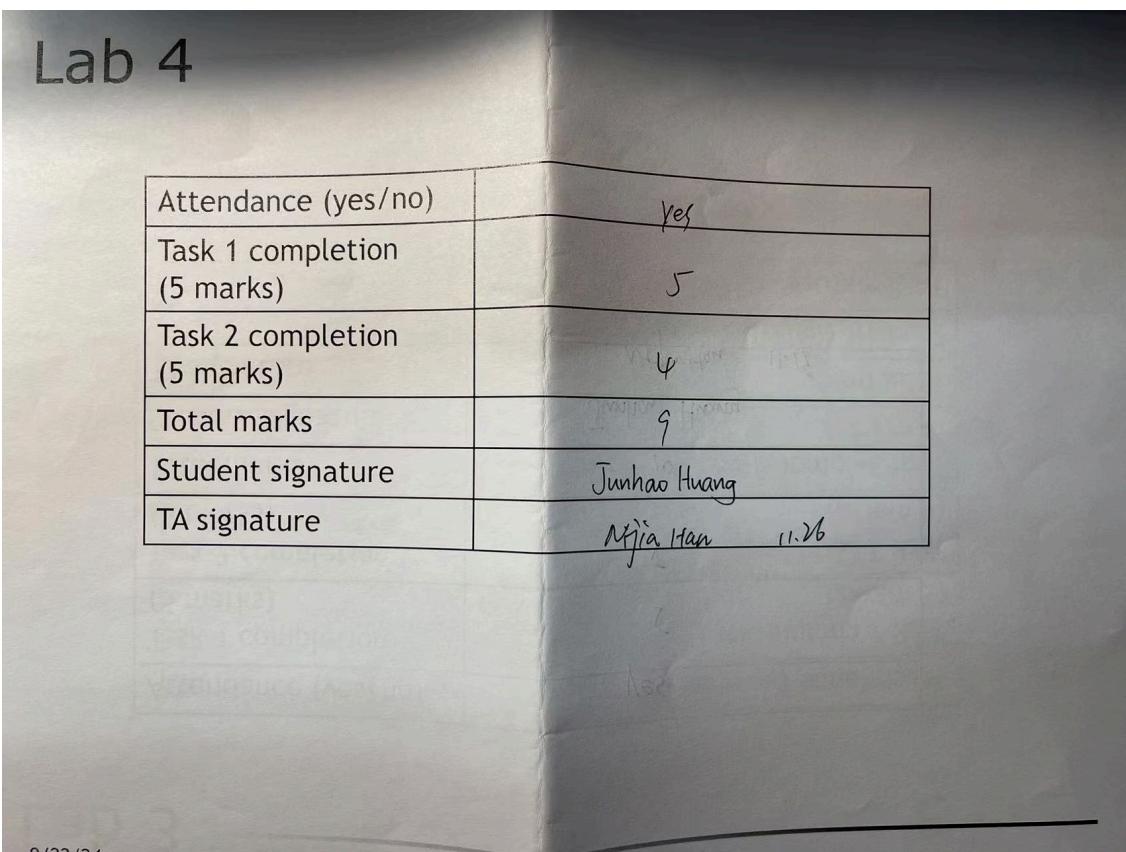
D:\Projects\Java_Projects\src\Lab3>
```

## SAX



```
C:\Windows\System32\cmd.exe + - Microsoft Windows [版本 10.0 22631.4460]
(c) Microsoft Corporation。保留所有权利。
D:\Projects\Java_Projects\src\Lab3>java SAXParserDemo
Roll No : 393
First Name: dinkar
Last Name: kad
Nick Name: dinkar
Marks: 85
End Element :student
Roll No : 493
First Name: Vaneet
Last Name: Gupta
Nick Name: vinni
Marks: 95
End Element :student
Roll No : 593
First Name: jasvir
Last Name: singn
Nick Name: jazz
Marks: 90
End Element :student
D:\Projects\Java_Projects\src\Lab3>
```

## Lab 4 KnowledgeGraph



Lab 4

Attendance (yes/no)	yes
Task 1 completion (5 marks)	5
Task 2 completion (5 marks)	4
Total marks	9
Student signature	Junhao Huang
TA signature	Njia Han 11.26

# Task 1

SPARQL   HTML5 table	
<b>athlete</b>	
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	
<a href="http://dbpedia.org/resource/Category:Cristiano_Ronaldo">http://dbpedia.org/resource/Category:Cristiano_Ronaldo</a>	

SPARQL   HTML5 table	
<b>athlete</b>	<b>place</b>
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Madeira">http://dbpedia.org/resource/Madeira</a>

athlete	place	cityName
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"Funchal"@en
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"فونشال"@ar
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"Funchal"@ca
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"Funchal"@cs
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"Funchal"@de
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"Фуншал"@el
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"Funchal"@eo
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"Funchal"@es
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"Funchal"@eu
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"Funchal"@frin
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"Funchal"@it
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"Funchal"@fr
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"ファンシャル"@ja
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"푼샬"@ko
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"Funchal"@nl
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"Funchal"@pt1
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"Funchal"@pt
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"Фуншал"@ru

athlete	place	cityName
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"Funchal"@en

## SPARQL | HTML5 table

athlete	place	cityName	region
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"Funchal"@en	"Madeira"@en

17:48  
2024/11/19

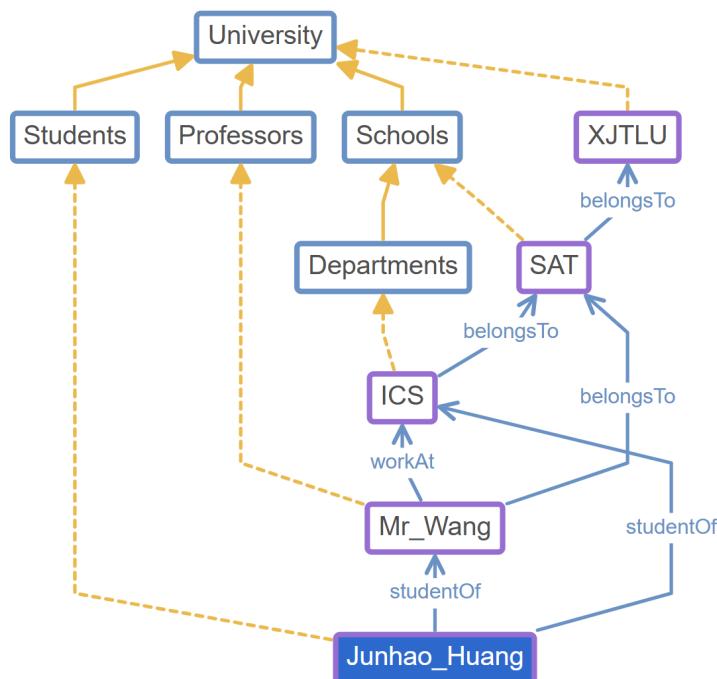
## SPARQL | HTML5 table

athlete	place	region
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"Madeira"@en

17:49  
2024/11/19

SPARQL   HTML5 table		
athlete	place	cityName
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Funchal">http://dbpedia.org/resource/Funchal</a>	"Funchal"@en
<a href="http://dbpedia.org/resource/Cristiano_Ronaldo">http://dbpedia.org/resource/Cristiano_Ronaldo</a>	<a href="http://dbpedia.org/resource/Madeira">http://dbpedia.org/resource/Madeira</a>	"Madeira"@en

## Task 2



The RDF code of the graph:

```

<?xml version="1.0"?>
<rdf:RDF xmlns="urn:webprotege:ontology:0043199f-c609-4775-9f0a-efbb33303640#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xm1="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:webprotege="http://webprotege.stanford.edu/"/>
<owl:Ontology rdf:about="urn:webprotege:ontology:0043199f-c609-4775-9f0a-efbb33303640#"/>
  
```

```

<!--

///////////
/////
// object Properties
//


/////////
-->

<!-- http://webprotege.stanford.edu/R9PoKwuk24PwAYyqbtMFIxu -->

<owl:ObjectProperty
rdf:about="http://webprotege.stanford.edu/R9PoKwuk24PwAYyqbtMFIxu">
    <rdfs:subPropertyof
rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
        <rdfs:label>workAt</rdfs:label>
    </owl:ObjectProperty>




<!-- http://webprotege.stanford.edu/RBKk1JzgB7nwMXJr8uXqvII -->

<owl:ObjectProperty
rdf:about="http://webprotege.stanford.edu/RBKk1JzgB7nwMXJr8uXqvII">
    <rdfs:subPropertyof
rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
        <rdfs:label>belongsTo</rdfs:label>
    </owl:ObjectProperty>




<!-- http://webprotege.stanford.edu/RCL5CAkBFBKX0Cw88JcRXHH -->

<owl:ObjectProperty
rdf:about="http://webprotege.stanford.edu/RCL5CAkBFBKX0Cw88JcRXHH">
    <rdfs:subPropertyof
rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
        <rdfs:label>studentOf</rdfs:label>
    </owl:ObjectProperty>




<!--

/////////
/////
//
```

```

// classes
//

///////////////////////////////
/////
-->

<!-- http://webprotege.stanford.edu/R9S46fKTuMhMBwY0WLnROQA -->

<owl:Class
  rdf:about="http://webprotege.stanford.edu/R9S46fKTuMhMBwY0WLnROQA">
    <rdfs:subClassOf
      rdf:resource="http://webprotege.stanford.edu/RCbv9PMZpvT7R0q3nuG32iy"/>
      <rdfs:label>Students</rdfs:label>
    </owl:Class>

<!-- http://webprotege.stanford.edu/RCbv9PMZpvT7R0q3nuG32iy -->

<owl:Class
  rdf:about="http://webprotege.stanford.edu/RCbv9PMZpvT7R0q3nuG32iy">
    <rdfs:label>University</rdfs:label>
  </owl:Class>

<!-- http://webprotege.stanford.edu/RDNNFK313i7wkhJFKj3rFR -->

<owl:Class
  rdf:about="http://webprotege.stanford.edu/RDNNFK313i7wkhJFKj3rFR">
    <rdfs:subClassOf
      rdf:resource="http://webprotege.stanford.edu/RhY1iQdak4T9rvRARJe5an"/>
      <rdfs:label>Departments</rdfs:label>
    </owl:Class>

<!-- http://webprotege.stanford.edu/RhY1iQdak4T9rvRARJe5an -->

<owl:Class  rdf:about="http://webprotege.stanford.edu/RhY1iQdak4T9rvRARJe5an">
  <rdfs:subClassOf
    rdf:resource="http://webprotege.stanford.edu/RCbv9PMZpvT7R0q3nuG32iy"/>
    <rdfs:label>Schools</rdfs:label>
  </owl:Class>

<!-- http://webprotege.stanford.edu/RrhWazT2CcY7C1J611CHQh -->

<owl:Class  rdf:about="http://webprotege.stanford.edu/RrhWazT2CcY7C1J611CHQh">
  <rdfs:subClassOf
    rdf:resource="http://webprotege.stanford.edu/RCbv9PMZpvT7R0q3nuG32iy"/>

```

```

<rdfs:label>Professors</rdfs:label>
</owl:Class>

<!--

//////////////////////////////  

/////  

//  

// Individuals  

//  

//////////////////////////////  

//////  

-->

<!-- http://webprotege.stanford.edu/R88U3jjQfvkgofovEef261U -->

<owl:NamedIndividual
rdf:about="http://webprotege.stanford.edu/R88U3jjQfvkgofovEef261U">
    <rdf:type
rdf:resource="http://webprotege.stanford.edu/RDNNFK313i7wkh0JFKj3rFR"/>
        <webprotege:RBKKlJzgB7nwMXJr8uXqvII
rdf:resource="http://webprotege.stanford.edu/R9a1823Uiuzo0BjfikJJHPE"/>
            <rdfs:label>ICS</rdfs:label>
</owl:NamedIndividual>

<!-- http://webprotege.stanford.edu/R9a1823Uiuzo0BjfikJJHPE -->

<owl:NamedIndividual
rdf:about="http://webprotege.stanford.edu/R9a1823Uiuzo0BjfikJJHPE">
    <rdf:type
rdf:resource="http://webprotege.stanford.edu/Rhy1iQdak4T9rvRARJe5an"/>
        <webprotege:RBKKlJzgB7nwMXJr8uXqvII
rdf:resource="http://webprotege.stanford.edu/RddkcUe6hwpzcyWJCXgufjx"/>
            <rdfs:label>SAT</rdfs:label>
</owl:NamedIndividual>

<!-- http://webprotege.stanford.edu/RBBDvwUUedRODwdOMy43uz4 -->

<owl:NamedIndividual
rdf:about="http://webprotege.stanford.edu/RBBDvwUUedRODwdOMy43uz4">
    <rdf:type
rdf:resource="http://webprotege.stanford.edu/RrhwazT2CcY7C1J611CHQh"/>
        <webprotege:R9PoKwuK24PwAYyqbtMFIxu
rdf:resource="http://webprotege.stanford.edu/R88U3jjQfvkgofovEef261U"/>
            <webprotege:RBKKlJzgB7nwMXJr8uXqvII
rdf:resource="http://webprotege.stanford.edu/R9a1823Uiuzo0BjfikJJHPE"/>
```

```
<rdfs:label>Mr_wang</rdfs:label>
</owl:NamedIndividual>

<!-- http://webprotege.stanford.edu/RDdkcUe6hwpzcyWjCXgufjX --&gt;

&lt;owl:NamedIndividual
rdf:about="http://webprotege.stanford.edu/RDdkcUe6hwpzcyWjCXgufjX"&gt;
&lt;rdf:type
rdf:resource="http://webprotege.stanford.edu/RCbv9PMZpvT7R0q3nug32iy"/&gt;
&lt;rdfs:label&gt;XJTLU&lt;/rdfs:label&gt;
&lt;/owl:NamedIndividual&gt;

<!-- http://webprotege.stanford.edu/REGNMENmTcCbocir7TVyL6 --&gt;

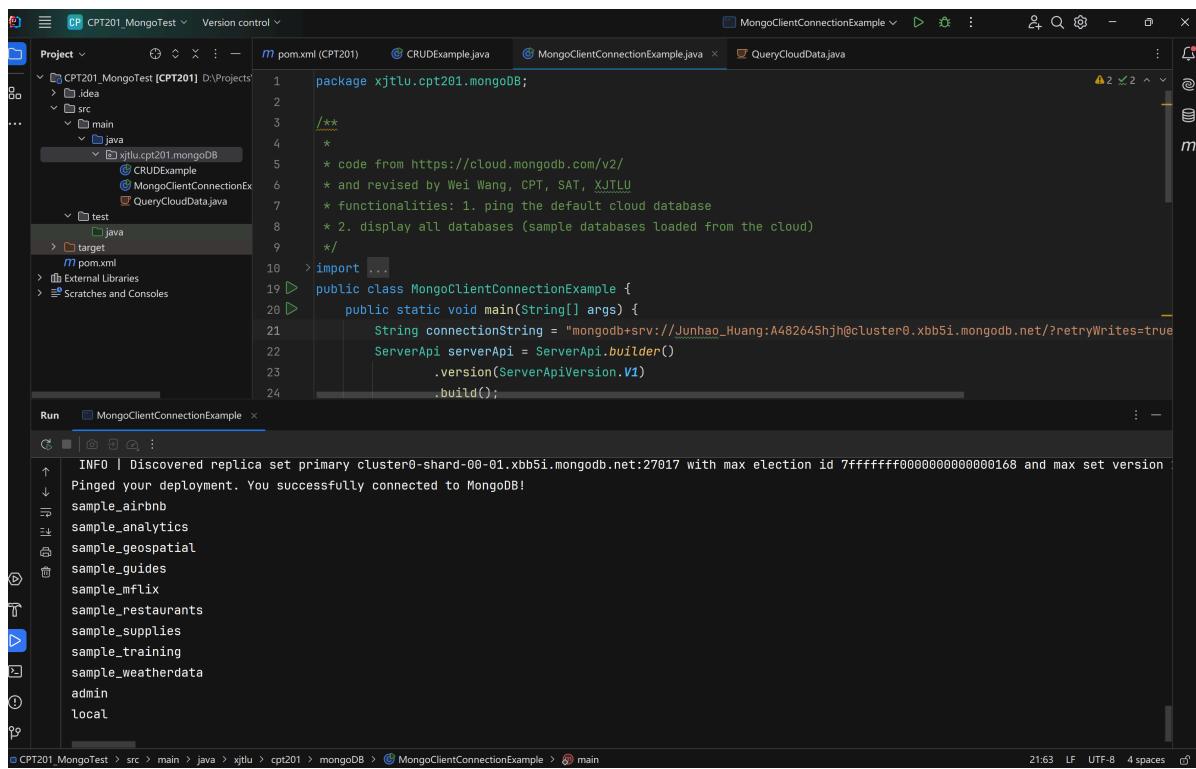
&lt;owl:NamedIndividual
rdf:about="http://webprotege.stanford.edu/REGNMENmTcCbocir7TVyL6"&gt;
&lt;rdf:type
rdf:resource="http://webprotege.stanford.edu/R9S46fKTuMhMBwY0WLnROQA"/&gt;
&lt;webprotege:RCL5CAkBFBKX0Cw88JcRXHH
rdf:resource="http://webprotege.stanford.edu/R88U3jjQfvkgofovEf261u"/&gt;
&lt;webprotege:RCL5CAkBFBKX0Cw88JcRXHH
rdf:resource="http://webprotege.stanford.edu/RBBDvwUUedRODwdOMy43uz4"/&gt;
&lt;rdfs:label&gt;Junhao_Huang&lt;/rdfs:label&gt;
&lt;/owl:NamedIndividual&gt;
&lt;/rdf:RDF&gt;

<!-- Generated by the OWL API (version 4.5.13) https://github.com/owlcs/owlapi --
&gt;</pre>
```

# Lab 5 Java and MongoDB

## Lab 5

Attendance (yes/no)	Yes
Task 1 completion (2 marks)	2
Task 2 completion (4 marks)	4
Task 3 completion (4 marks)	4
Total marks	10
Student signature	Junhao Huang
TA signature	Zimn Wang 11.26



The screenshot shows an IDE interface with the following details:

- Project Structure:** The project is named "CPT201.MongoTest". It contains a "src" directory with "main" and "test" sub-directories. Under "main/java", there is a package named "xjtlu.cpt201.mongoDB" which contains three files: "CRUDExample.java", "MongoClientConnectionExample.java", and "QueryCloudData.java". The "MongoClientConnectionExample.java" file is currently selected and shown in the code editor.
- Code Editor:** The code for "MongoClientConnectionExample.java" is displayed. It imports the necessary MongoDB Java API classes and defines a main method that connects to a MongoDB instance using a provided connection string.
- Run Tab:** The "Run" tab is active, showing the selected configuration is "MongoClientConnectionExample".
- Terminal:** The terminal window shows the output of a ping command to the MongoDB deployment, indicating successful connection.
- Bottom Status Bar:** The status bar displays the path "CPT201.MongoTest > src > main > java > xjtlu > cpt201 > mongoDB > MongoClientConnectionExample > main", along with the current time "21:33", line separator "LF", encoding "UTF-8", and code style "4 spaces".

The screenshot shows an IDE interface with the following details:

- Project:** CPT201\_MongoTest [CPT201]
- File:** CRUDEExample.java
- Code Content:** Java code demonstrating MongoDB CRUD operations using the MongoClient API.

```
public class CRUDEExample {
    public static void main(String[] args) {
        // update data
        Document query = new Document();
        query.put("name", "Leo Messi");
        Document newDocument = new Document();
        newDocument.put("company", "Maimi FC");
        Document updateObject = new Document();
        updateObject.put("$set", newDocument);
        collection.updateOne(query, updateObject);
        System.out.println("The record has been updated.");
        // read data
        Document searchQuery = new Document();
        searchQuery.put("name", "Leo Messi");
    }
}
```

- Run Output:** The terminal window shows the execution of the code and the resulting output.

```
INFO | Monitor thread successfully connected to server with description ServerDescription{address=cluster0-shard-00-00.xbb5i.mongodb.net:27017, type=ReplicaSet, id=7ff...}
INFO | Monitor thread successfully connected to server with description ServerDescription{address=cluster0-shard-00-02.xbb5i.mongodb.net:27017, type=ReplicaSet}
INFO | Monitor thread successfully connected to server with description ServerDescription{address=cluster0-shard-00-01.xbb5i.mongodb.net:27017, type=ReplicaSet}
INFO | Discovered replica set primary cluster0-shard-00-01.xbb5i.mongodb.net:27017 with max election id 7fffffff00000000000000168 and max set version 1
myMongoDb contains the following collections.
customers
The record Document{{name=Leo Messi, company=Barcelona FC, _id=6745a2d205bfb11d602a8d69}} is inserted to collection com.mongodb.client.internal.MongoCollection<Document>
The record has been updated.
Document{{_id=6745a2d205bfb11d602a8d69, name=Leo Messi, company=Maimi FC}}
The record has been deleted.
The record has been deleted.

Process finished with exit code 0
```

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project:** CPT201\_MongoTest [CPT201]
- Pom.xml:** pom.xml (CPT201)
- Source Code:** MongoClientConnectionExample.java, CRUDEExample.java, QueryCloudData.java
- Code Editor:** The QueryCloudData.java file is open, showing code to query a MongoDB collection for documents where the name is "Private Room in Bushwick".
- Run Tab:** Shows the command run: D:\Development\Java\Java-17\bin\java.exe -javaagent:D:\Development\IntelliJ\_IDEA\lib\idea\_rt.jar=17161:D:\Development\IntelliJ\_IDEA\bin -Dfile.encoding=UTF-8.
- Terminal Output:** Displays the execution of the Java application, showing the connection to a MongoDB cluster and the retrieval of a document matching the query.

# Lab 6 XML and MongoDB

The screenshot shows the phpMyAdmin interface for a MySQL database named 'junhaohuang2202'. The 'books' table is selected, displaying 12 rows of book information. The columns are id, book\_id, author, title, genre, price, publish\_date, and description. The data includes various books like 'XML Developer's Guide', 'Midnight Rain', and 'The Sundered Grail'.

	id	book_id	author	title	genre	price	publish_date	description
1	bk101	Gambardella, Matthew	XML Developer's Guide	Computer	44.95	2000-10-01	An in-depth look at creating applications w...	
2	bk102	Ralls, Kim	Midnight Rain	Fantasy	5.95	2000-12-16	A former architect battles corporate zombies, ...	
3	bk103	Corets, Eva	Maeve Ascendant	Fantasy	5.95	2000-11-17	After the collapse of a nanotechnology soci...	
4	bk104	Corets, Eva	Oberon's Legacy	Fantasy	5.95	2001-03-10	In post-apocalypse England, the mysterious ...	
5	bk105	Corets, Eva	The Sundered Grail	Fantasy	5.95	2001-09-10	The two daughters of Maeve, half-sisters, b...	
6	bk106	Randall, Cynthia	Lover Birds	Romance	4.95	2000-09-02	When Carla meets Paul at an ornithology con...	
7	bk107	Thurman, Paula	Splish Splash	Romance	4.95	2000-11-02	A deep sea diver finds true love twenty tho...	
8	bk108	Knorr, Stefan	Creepy Crawlies	Horror	4.95	2000-12-06	An anthology of horror stories about roaches, ...	
9	bk109	Kress, Peter	Paradox Lost	Science Fiction	6.95	2000-11-02	After an inadvertent trip through a Heisenberg ...	
10	bk110	O'Brien, Tim	Microsoft .NET: The Programming Bible	Computer	36.95	2000-12-09	Microsoft's .NET initiative is explored in ...	
11	bk111	O'Brien, Tim	MSXML3: A Comprehensive Guide	Computer	36.95	2000-12-01	The Microsoft MSXML3 parser is covered in d...	
12	bk112	Galos, Mike	Visual Studio 7: A Comprehensive Guide	Computer	49.95	2001-04-16	Microsoft Visual Studio 7 is explored in depth, ...	

The screenshot shows an IntelliJ IDEA project named 'CPT201'. The 'src' directory contains a 'Lab6' package with a 'LabSix.java' file. The code uses Java to read from 'books.xml' and interact with a database. The terminal window shows the output of running the code, which successfully inserts values into the database.

```
public class LabSix {
    public void databaseOperation(ArrayList<LinkedHashMap<String, Object>> list) {
        //close database connections.
        // rs.close();
        conn.close();

    } catch (ClassNotFoundException e) {
        System.out.println("Cannot find JDBC Driver!");
    }
}
```

Output in the terminal:

```
D:\Development\Java\Java-17\bin\java.exe -javaagent:D:\Development\IntelliJ_IDEA\lib\idea_rt.jar=38657:D:\Development\IntelliJ_IDEA\bin -Dfile.encoding=UTF-8
Root element :catalog
Succeeded connecting to the Database.
Start to insert values into the database
Rows affected: 1
End query.
End query without an index.
```

The Code of the In Class Test:

The code can successfully get the result below

```
package Lab6;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import javax.xml.parsers.DocumentBuilder;
```

```

import javax.xml.parsers.DocumentBuilderFactory;
import java.io.File;
import java.sql.*;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.ArrayList;
import java.util.LinkedHashMap;
public class LabSix {
    // initiate database information
    static final String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";
    //replace the following three parameter values with your own ones
    static final String DB_URL = "jdbc:mysql://10.7.1.127/junhaohuang2202";
    static final String USER = "JunhaoHuang2202";
    static final String PASS = "123";
    public ArrayList<LinkedHashMap<String, Object>> extractInfo(String filepath)
    {
        ArrayList<LinkedHashMap<String, Object>> list = new ArrayList<>();
        try {
            File inputFile = new File("src/Lab6/books.xml");
            DocumentBuilderFactory dbFactory =
                DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
            Document doc = dBuilder.parse(inputFile);
            doc.getDocumentElement().normalize();
            System.out.println("Root element :" +
                doc.getDocumentElement().getNodeName());
            NodeList nList = doc.getElementsByTagName("book");
            // get each book info
            for (int temp = 0; temp < nList.getLength(); temp++) {
                LinkedHashMap<String, Object> entity = new LinkedHashMap<>();
                Node nNode = nList.item(temp);
                // print book
                // System.out.println("\nCurrent Element:" + nNode.getNodeName()
                + temp);
                if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) nNode;
                    // get book_id
                    String book_id = eElement.getAttribute("id");
                    // get author
                    String author =
eElement.getElementsByTagName("author").item(0).getTextContent();
                    // get title
                    String title =
eElement.getElementsByTagName("title").item(0).getTextContent();
                    // get genre
                    String genre =
eElement.getElementsByTagName("genre").item(0).getTextContent();
                    // get price
                    String price =
eElement.getElementsByTagName("price").item(0).getTextContent();

```

```

        // get date
        String date =
    eElement.getElementsByTagName("publish_date").item(0).getTextContent();
        // get description
        String description =
    eElement.getElementsByTagName("description").item(0).getTextContent();
        // add to map
        entity.put("book_id", book_id);
        entity.put("author", author);
        entity.put("title", title);
        entity.put("genre", genre);
        entity.put("price", price);
        entity.put("date", date);
        entity.put("description", description);
        // 将当前的 entity 添加到 arraylist 当中
        list.add(entity);
    }
}
} catch (Exception e) {
    e.printStackTrace();
}
return list;
}

public void databaseOperation(ArrayList<LinkedHashMap<String, Object>>
    list){
    Connection conn;
    Statement stmt;
    ResultSet rs;
    String sql;
    String first;
    long begin;
    long end;
    try {
        Class.forName(JDBC_DRIVER);
        conn = DriverManager.getConnection(DB_URL, USER, PASS);
        if (!conn.isClosed()) {
            System.out.println("Succeeded connecting to the Database.");
        }
        stmt = conn.createStatement();
        //prepare query without an index
        System.out.println("Start to insert values into the database");
        for (int i = 0; i < list.size(); i++) {
            LinkedHashMap<String, Object> entity = list.get(i);
            try {
                String priceStr = (String) entity.get("price");
                String dateStr = (String) entity.get("date");
                String bookId = (String) entity.get("book_id");
                String author = (String) entity.get("author");
                String title = (String) entity.get("title");
                String genre = (String) entity.get("genre");
                String description = (String) entity.get("description");
                // check null value
                if (priceStr == null || dateStr == null || bookId == null ||

```

```
        author == null || title == null || genre == null ||
description == null) {
    System.out.println("Skipping incomplete data: " +
                       entity);
    continue;
}
// convert float and date
float price = Float.parseFloat(priceStr);
DateTimeFormatter formatter =
    DateTimeFormatter.ofPattern("yyyy-MM-dd");
LocalDate localDate = LocalDate.parse(dateStr, formatter);
java.sql.Date sqlDate = java.sql.Date.valueOf(localDate);
// 构建 SQL
sql = "INSERT INTO books (book_id, author, title, genre,
price, publish_date, description) VALUES (?, ?, ?, ?, ?, ?, ?,
?)";
try (PreparedStatement pstmt = conn.prepareStatement(sql))
{
    pstmt.setString(1, bookId);
    pstmt.setString(2, author);
    pstmt.setString(3, title);
    pstmt.setString(4, genre);
    pstmt.setFloat(5, price);
    pstmt.setDate(6, sqlDate);
    pstmt.setString(7, description);
    int affectedRows = pstmt.executeUpdate();
    System.out.println("Rows affected: " + affectedRows);
}
} catch (SQLException e) {
    System.err.println("SQL error: " + e.getMessage());
    e.printStackTrace();
} catch (Exception e) {
    System.err.println("Error: " + e.getMessage());
    e.printStackTrace();
}
}
System.out.println("End query.");
conn.close();
//close database connections.
conn.close();
} catch (ClassNotFoundException e) {
    System.out.println("Cannot find JDBC Driver!");
    e.printStackTrace();
} catch (SQLException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}
}
public static void main(String[] args) {
    Labsix labsix = new Labsix();
    ArrayList<LinkedHashMap<String, Object>> result =
        labsix.extractInfo("books.xml");
    for (LinkedHashMap<String, Object> entity : result) {
        System.out.printf((String) entity.get("book_id"));
    }
}
```

```
    tabsix.databaseOperation(result);
}
}
```