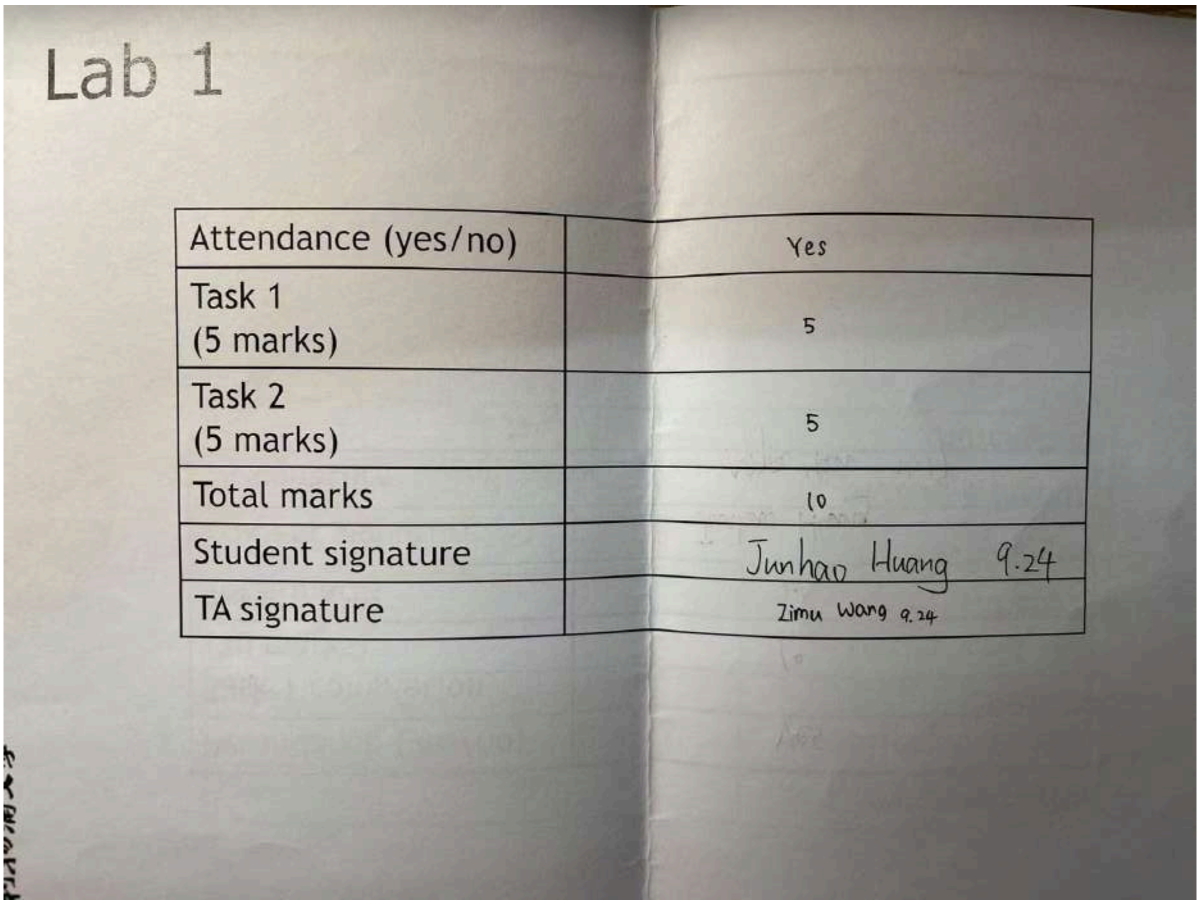
CPT201 Database Development and Design Lab Report

Name: Junhao Huang

Student ID: 2256792

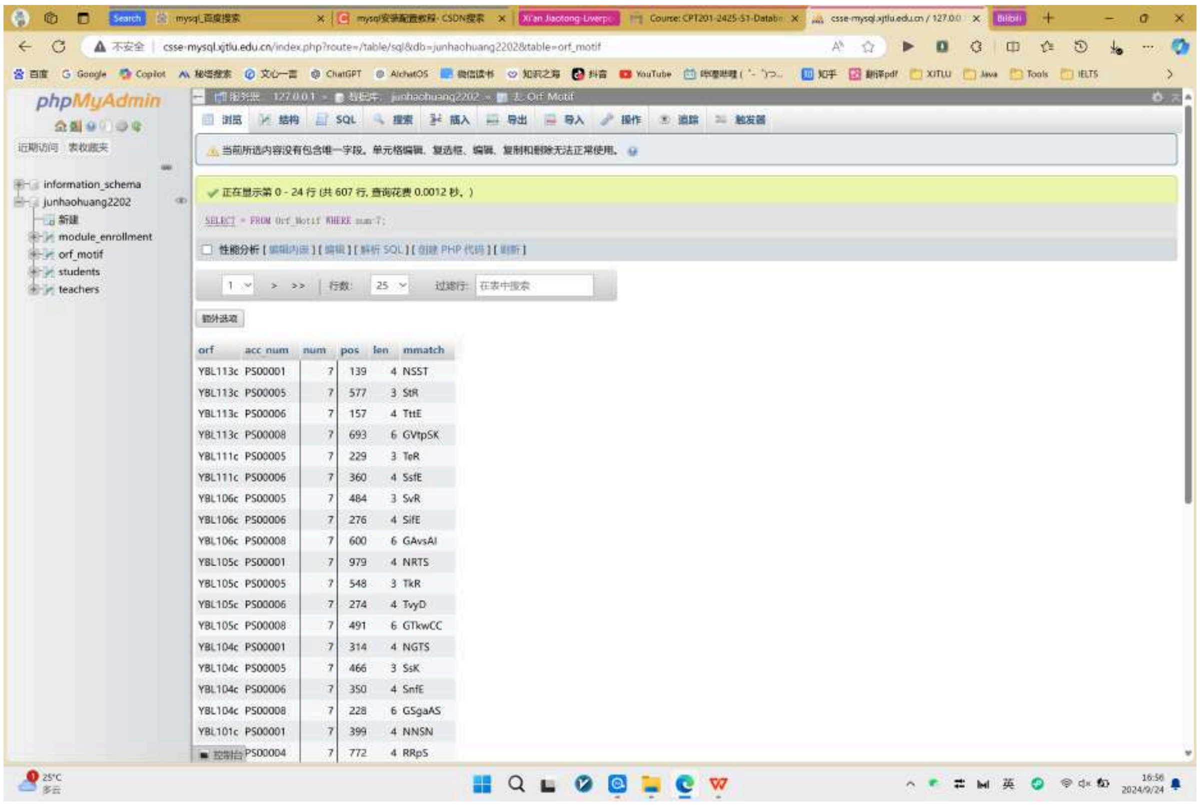
Date: 2024.12.12

Lab 1 JDBC



Lab 1

Attendance (yes/no)	Yes
Task 1 (5 marks)	5
Task 2 (5 marks)	5
Total marks	10
Student signature	Junhao Huang 9.24
TA signature	Zimu Wang 9.24



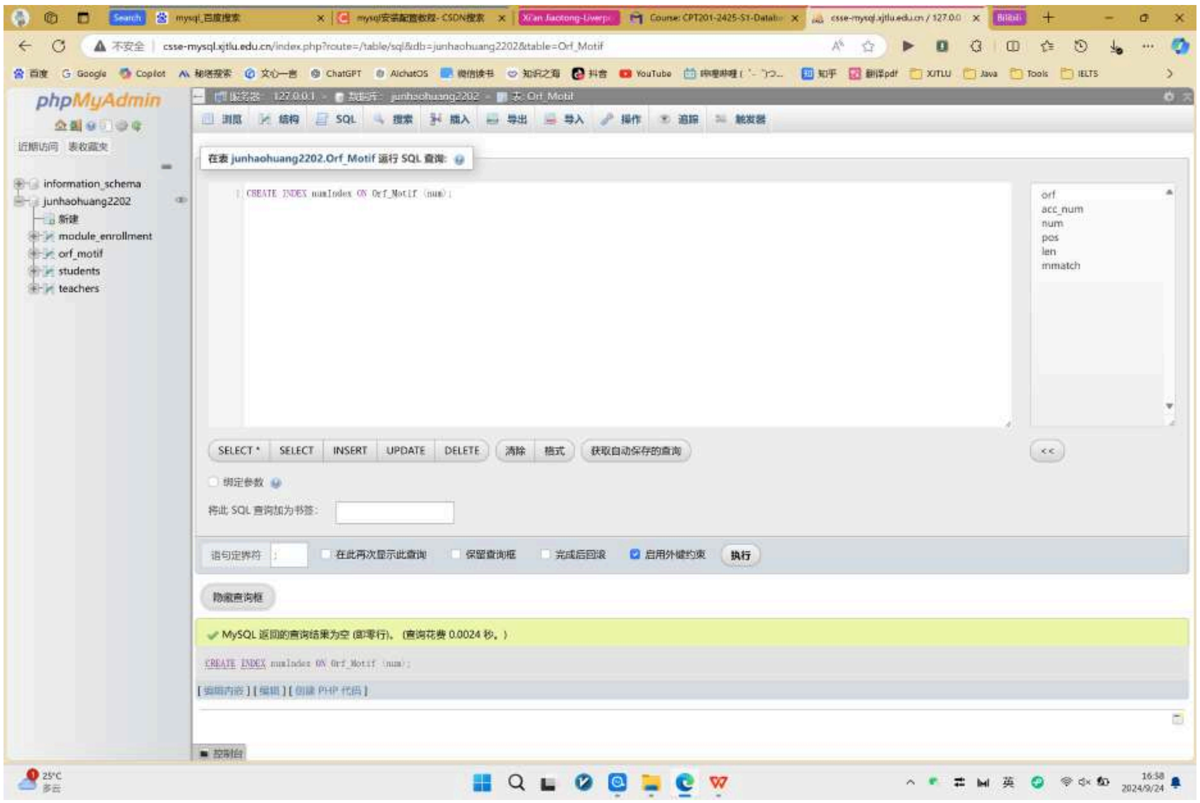
csse-mysql.xjtlu.edu.cn/index.php?route=/table/sql&db=junhaohuang2202&table=orf_motif

当前所选内容没有包含唯一字段。单元格编辑、复选框、编辑、复制和删除无法正常使用。

正在显示第 0 - 24 行 (共 607 行, 查询花费 0.0012 秒。)

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

orf	acc_num	num	pos	len	mmatch
YBL113c	PS00001	7	139	4	NSST
YBL113c	PS00005	7	577	3	StR
YBL113c	PS00006	7	157	4	TttE
YBL113c	PS00008	7	693	6	GVtpSK
YBL111c	PS00005	7	229	3	TeR
YBL111c	PS00006	7	360	4	SsfE
YBL106c	PS00005	7	484	3	SvR
YBL106c	PS00006	7	276	4	SifE
YBL106c	PS00008	7	600	6	GAvsAI
YBL105c	PS00001	7	979	4	NRTS
YBL105c	PS00005	7	548	3	TkR
YBL105c	PS00006	7	274	4	TvyD
YBL105c	PS00008	7	491	6	GTkwCC
YBL104c	PS00001	7	314	4	NGTS
YBL104c	PS00005	7	466	3	SsK
YBL104c	PS00006	7	350	4	SnfE
YBL104c	PS00008	7	228	6	GSgaAS
YBL101c	PS00001	7	399	4	NNSN
[illegible]	PS00004	7	772	4	RRpS



在表 junhaohuang2202.Orf_Motif 运行 SQL 查询:

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

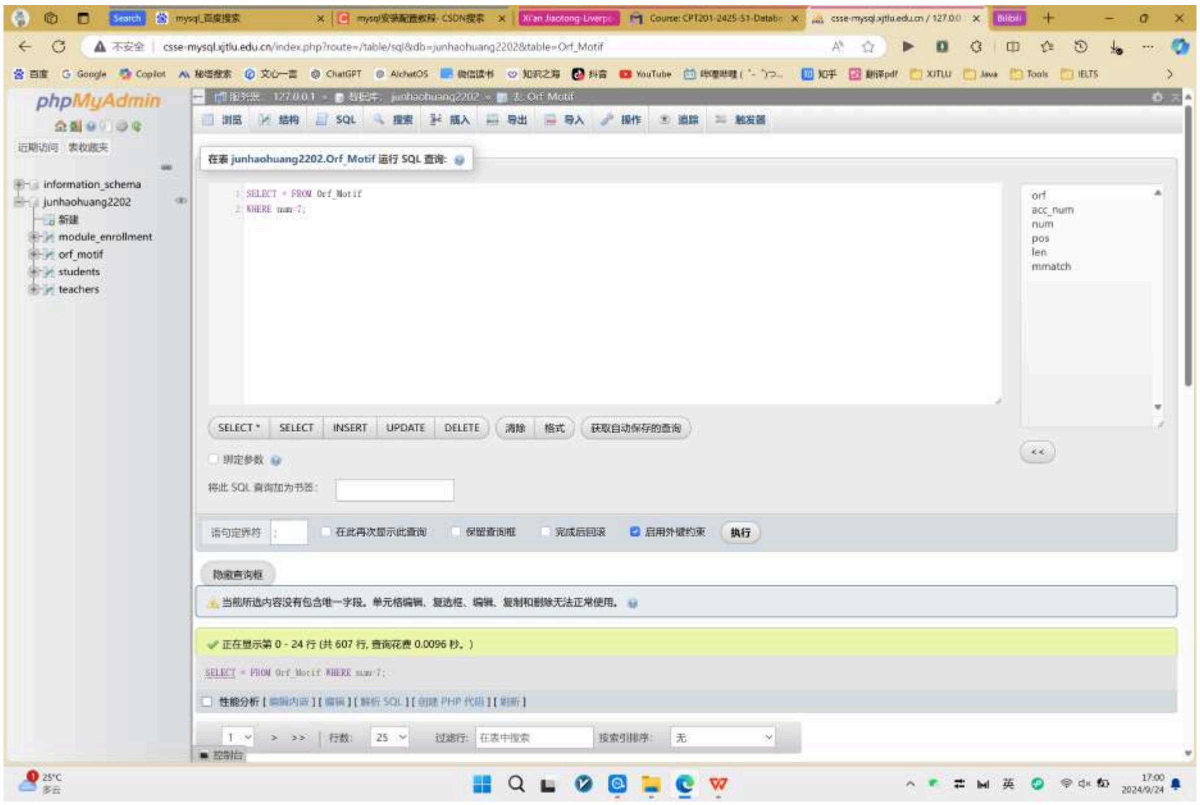
orf, acc_num, num, pos, len, mmatch

语句定界符 ; ☐ 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

✓ 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]



在表 junhaohuang2202.Orf_Motif 运行 SQL 查询:

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

orf, acc_num, num, pos, len, mmatch

语句定界符 ; ☐ 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

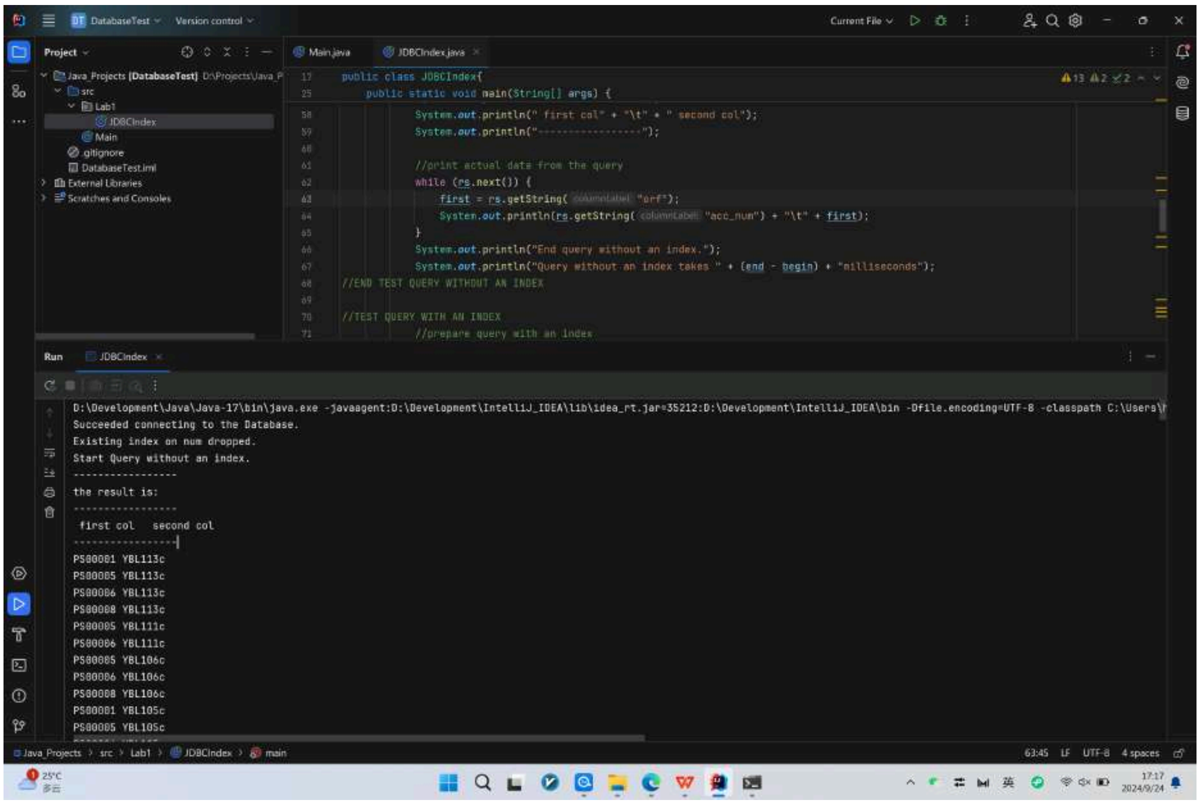
⚠ 当前所选内容没有包含唯一字段。单元格编辑、复选框、编辑、复制和删除无法正常使用。

✓ 正在显示第 0 - 24 行 (共 607 行, 查询花费 0.0096 秒。)

```
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]

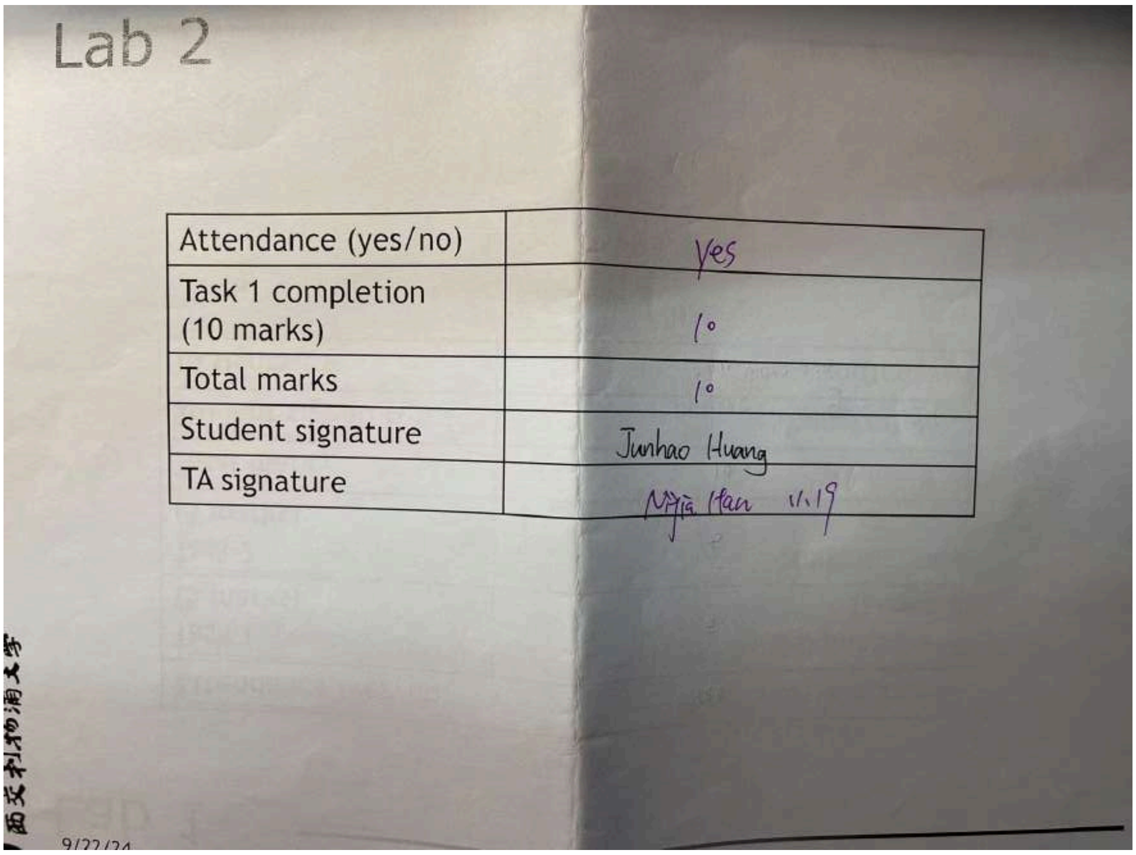
1 > >> | 行数: 25 过滤行: 在表中搜索 按索引排序: 无



```java
public class JDBCIndex{
    public static void main(String[] args) {
                System.out.println(" first col" + "\t" + " second col");
                System.out.println("-----------------");

                //print actual data from the query
                while (rs.next()) {
                    first = rs.getString( "orf");
                    System.out.println(rs.getString( "acc_num") + "\t" + first);
                }
                System.out.println("End query without an index.");
                System.out.println("Query without an index takes " + (end - begin) + "milliseconds");
    //END TEST QUERY WITHOUT AN INDEX

    //TEST QUERY WITH AN INDEX
                //prepare query with an index
```

```
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\
Succeeded connecting to the Database.
Existing 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
```

Lab2 JTA

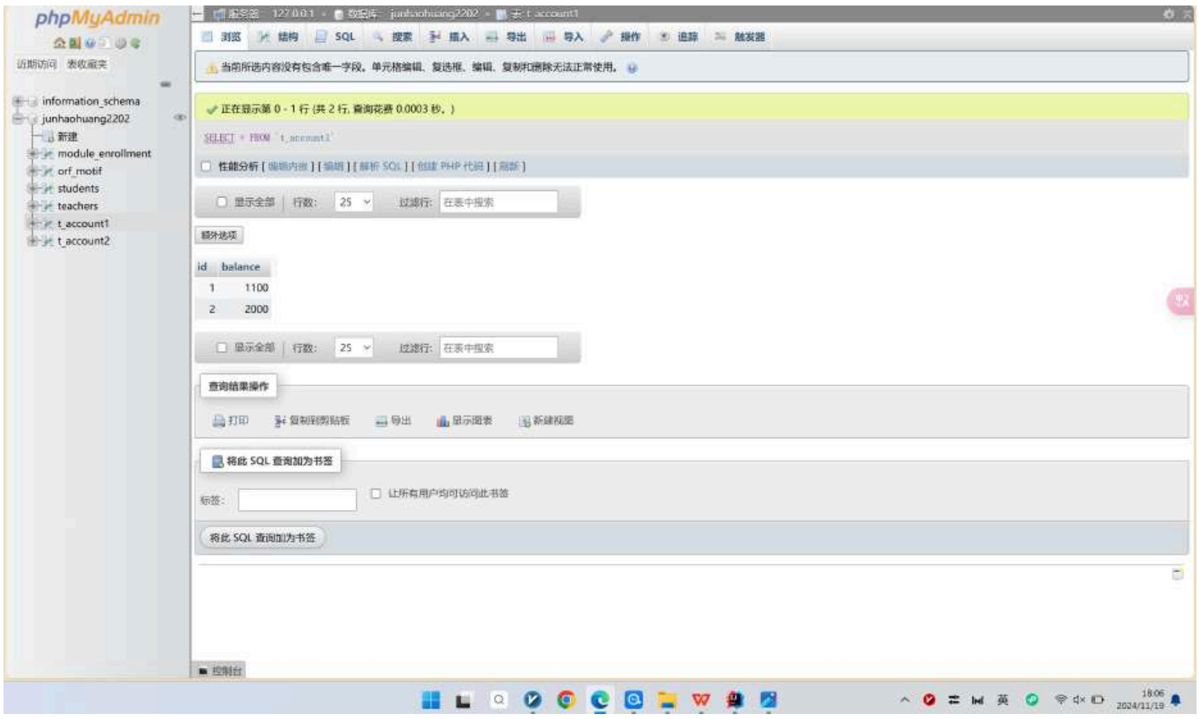


Lab 2

Attendance (yes/no)	yes
Task 1 completion (10 marks)	10
Total marks	10
Student signature	Junhao Huang
TA signature	[illegible] 11.19

西交利物浦大学

9/22/24



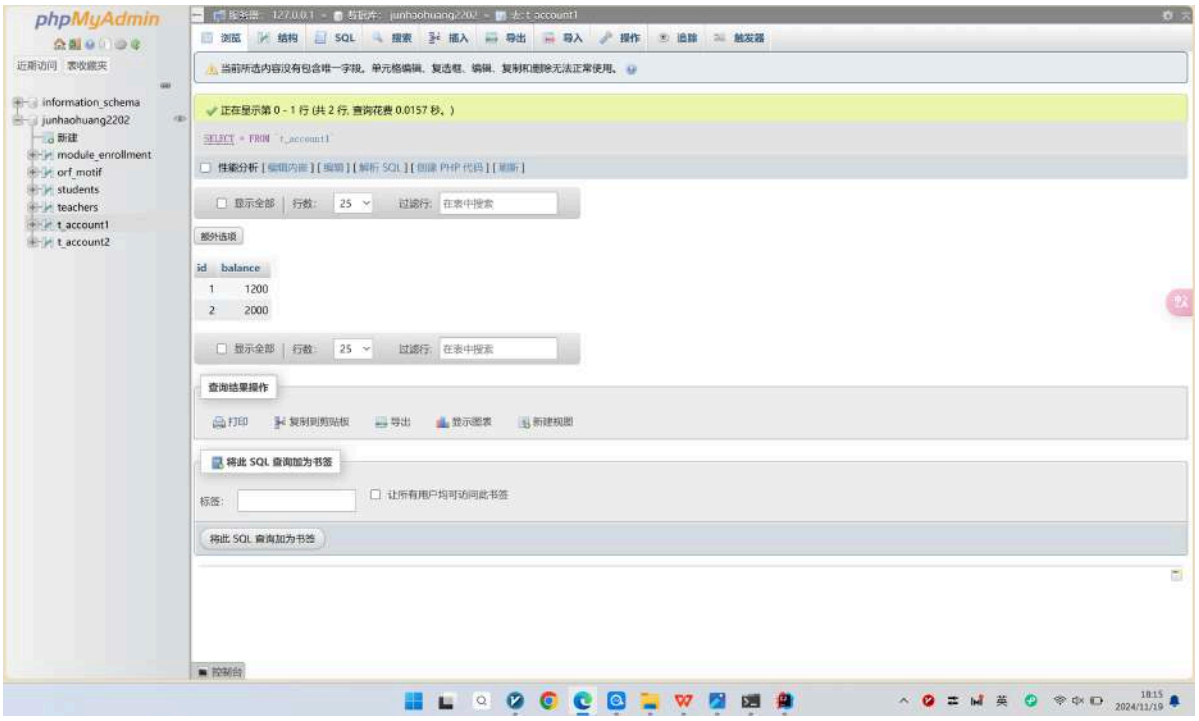
当前所选内容没有包含唯一字段。单元格编辑、复选框、编辑、复制和删除无法正常使用。

正在显示第 0 - 1 行 (共 2 行, 查询花费 0.0003 秒。)

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

id	balance
1	1100
2	2000

18:06
2024/11/19



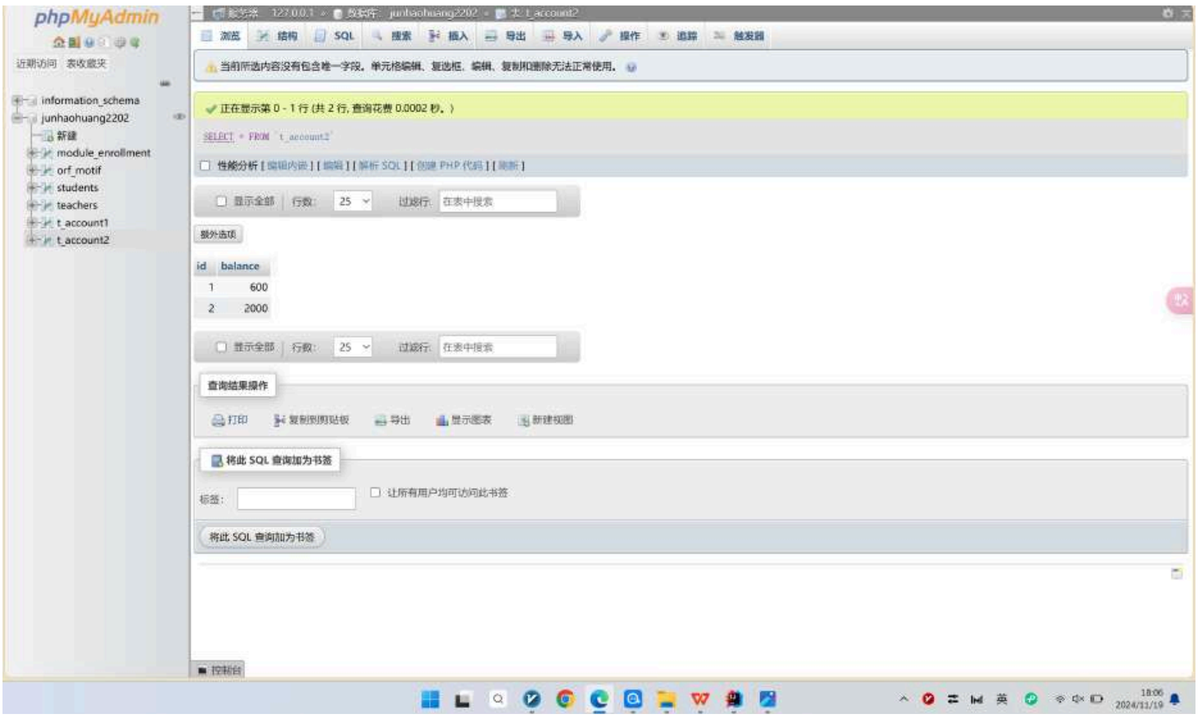
当前所选内容没有包含唯一字段。单元格编辑、复选框、编辑、复制和删除无法正常使用。

正在显示第 0 - 1 行 (共 2 行, 查询花费 0.0157 秒。)

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

id	balance
1	1200
2	2000

18:15
2024/11/19

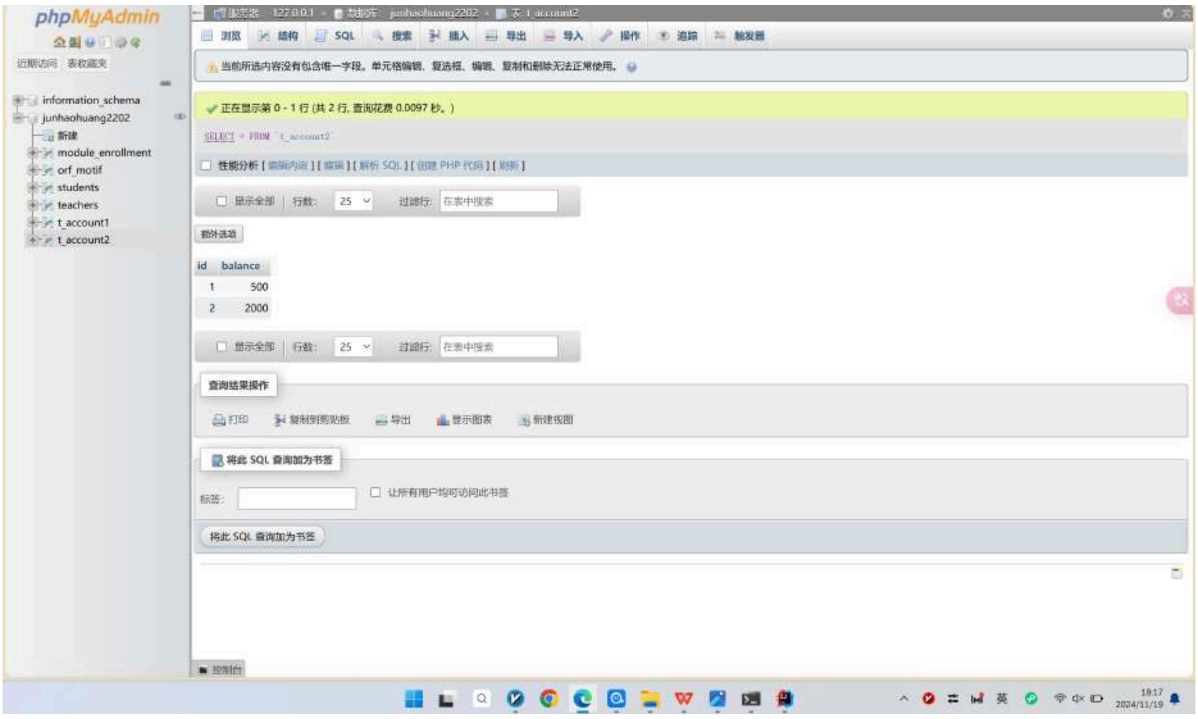

phpMyAdmin
近期访问 表收藏夹
information_schema
junhaohuang2202
新建
module_enrollment
orf_motif
students
teachers
t_account1
t_account2
服务器: 127.0.0.1 » 数据库: junhaohuang2202 » 表: t_account2
浏览 结构 SQL 搜索 插入 导出 导入 操作 追踪 触发器
性能分析 [编辑内嵌] [编辑] [解析 SQL] [创建 PHP 代码] [刷新]
显示全部 行数: 25 过滤行: 在表中搜索
额外选项
查询结果操作
打印 复制到剪贴板 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
控制台

当前所选内容没有包含唯一字段。单元格编辑、复选框、编辑、复制和删除无法正常使用。

正在显示第 0 - 1 行 (共 2 行, 查询花费 0.0002 秒。)

```sql
SELECT * FROM `t_account2`
```

id	balance
1	600
2	2000

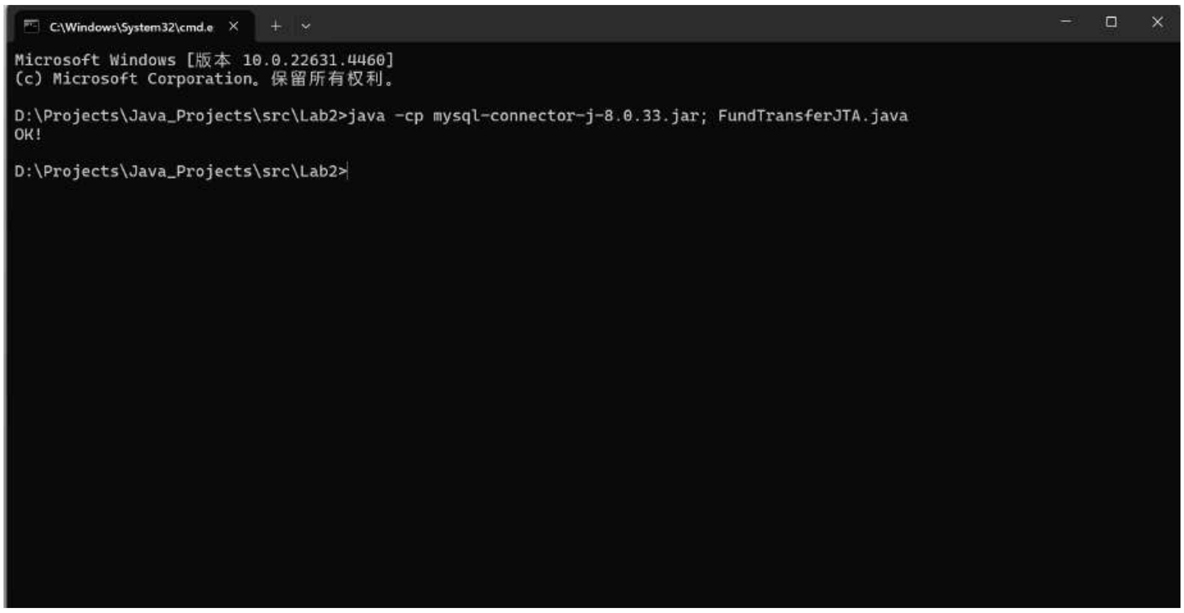

phpMyAdmin
近期访问 表收藏夹
information_schema
junhaohuang2202
新建
module_enrollment
orf_motif
students
teachers
t_account1
t_account2
服务器: 127.0.0.1 » 数据库: junhaohuang2202 » 表: t_account2
浏览 结构 SQL 搜索 插入 导出 导入 操作 追踪 触发器
性能分析 [编辑内嵌] [编辑] [解析 SQL] [创建 PHP 代码] [刷新]
显示全部 行数: 25 过滤行: 在表中搜索
额外选项
查询结果操作
打印 复制到剪贴板 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
控制台

当前所选内容没有包含唯一字段。单元格编辑、复选框、编辑、复制和删除无法正常使用。

正在显示第 0 - 1 行 (共 2 行, 查询花费 0.0097 秒。)

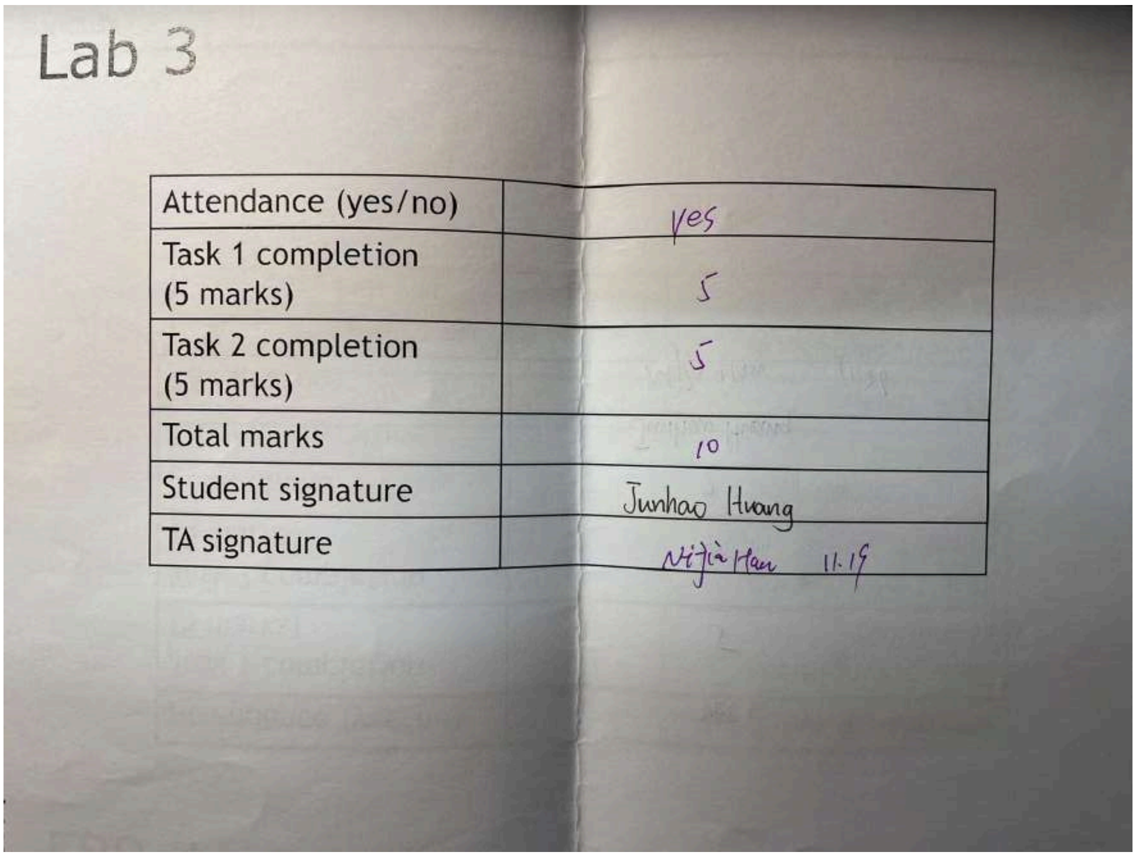
```sql
SELECT * FROM `t_account2`
```

id	balance
1	500
2	2000



```
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>
```

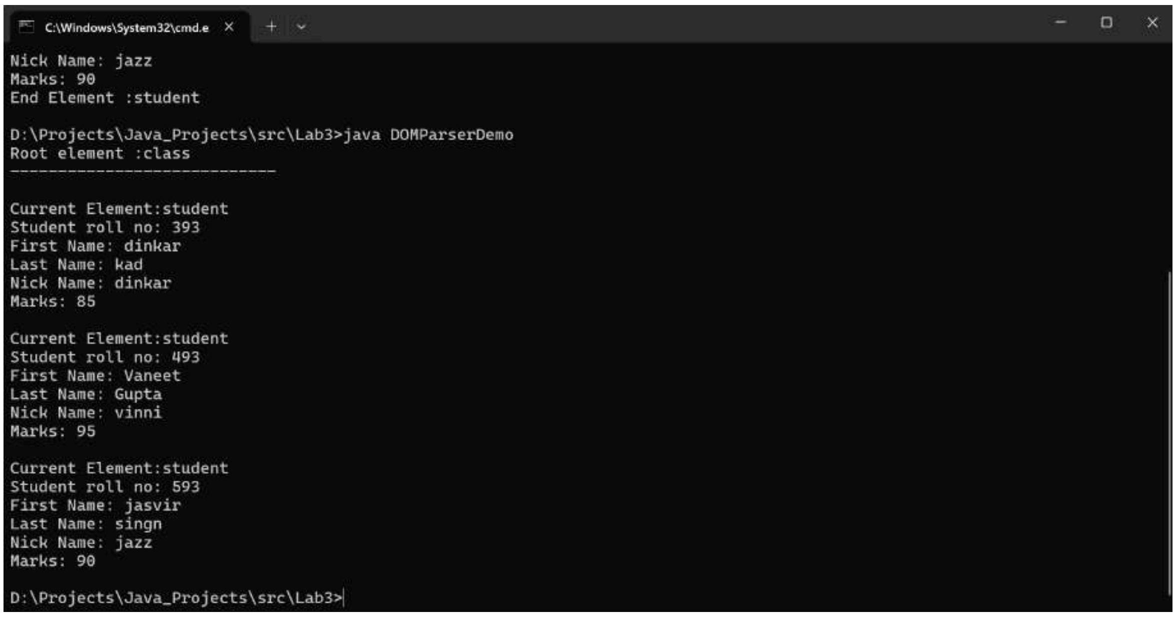
Lab 3 XML Processing



Lab 3

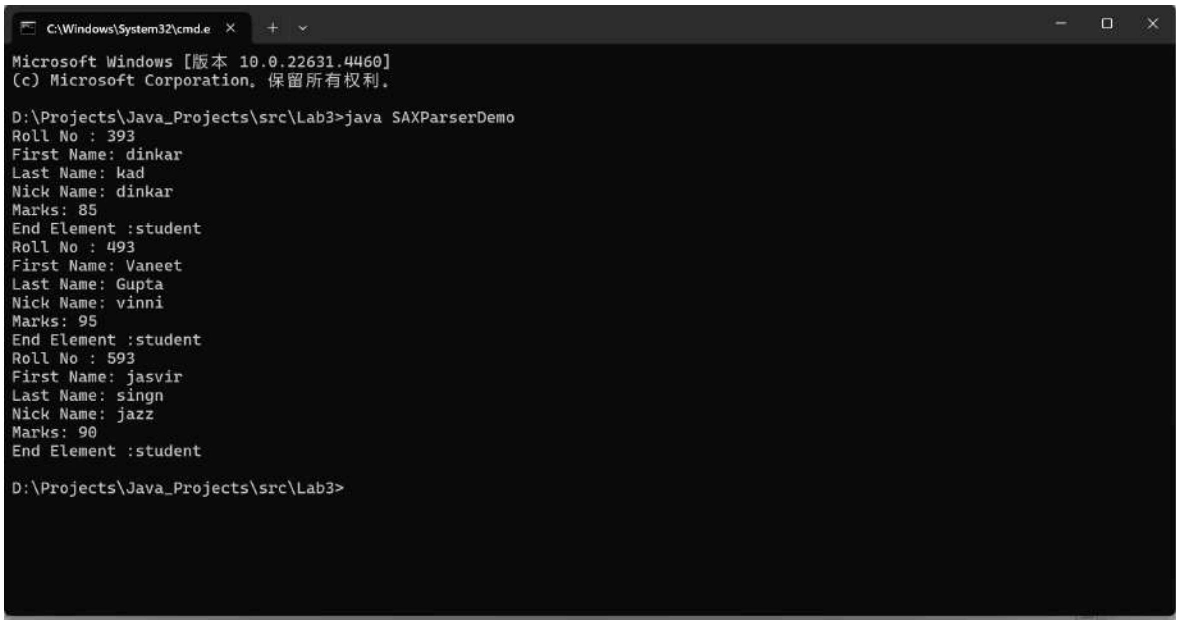
Attendance (yes/no)	yes
Task 1 completion (5 marks)	5
Task 2 completion (5 marks)	5
Total marks	10
Student signature	Junhao Huang
TA signature	[illegible] 11.19

DOM



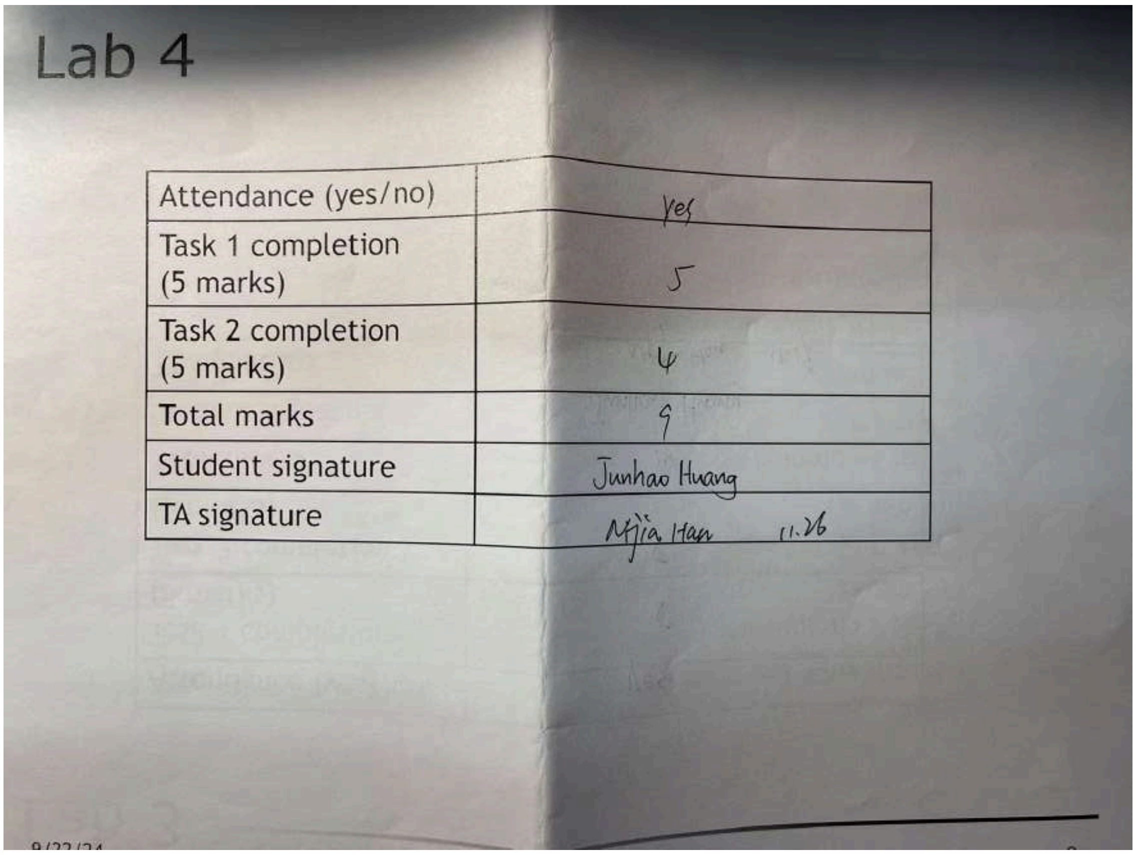
```
Nick Name: jazz
Marks: 90
End Element :student

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



```
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>
```

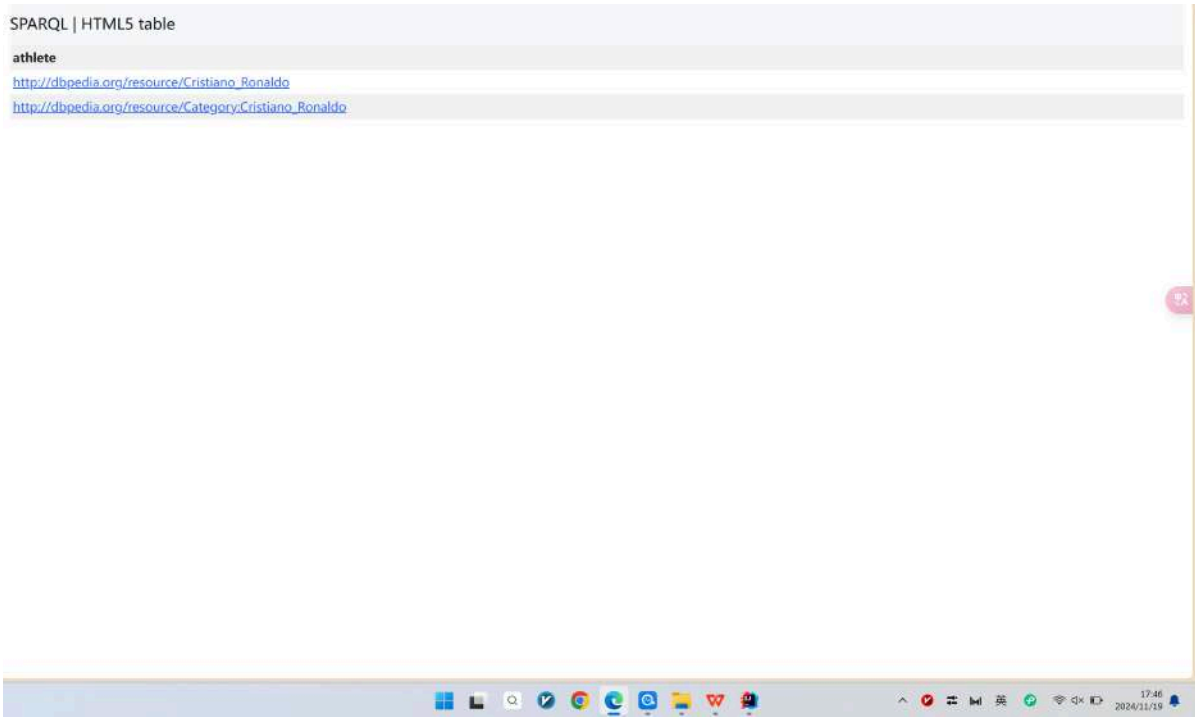
Lab 4 KnowledgeGraph



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	Mjia Han 11.26

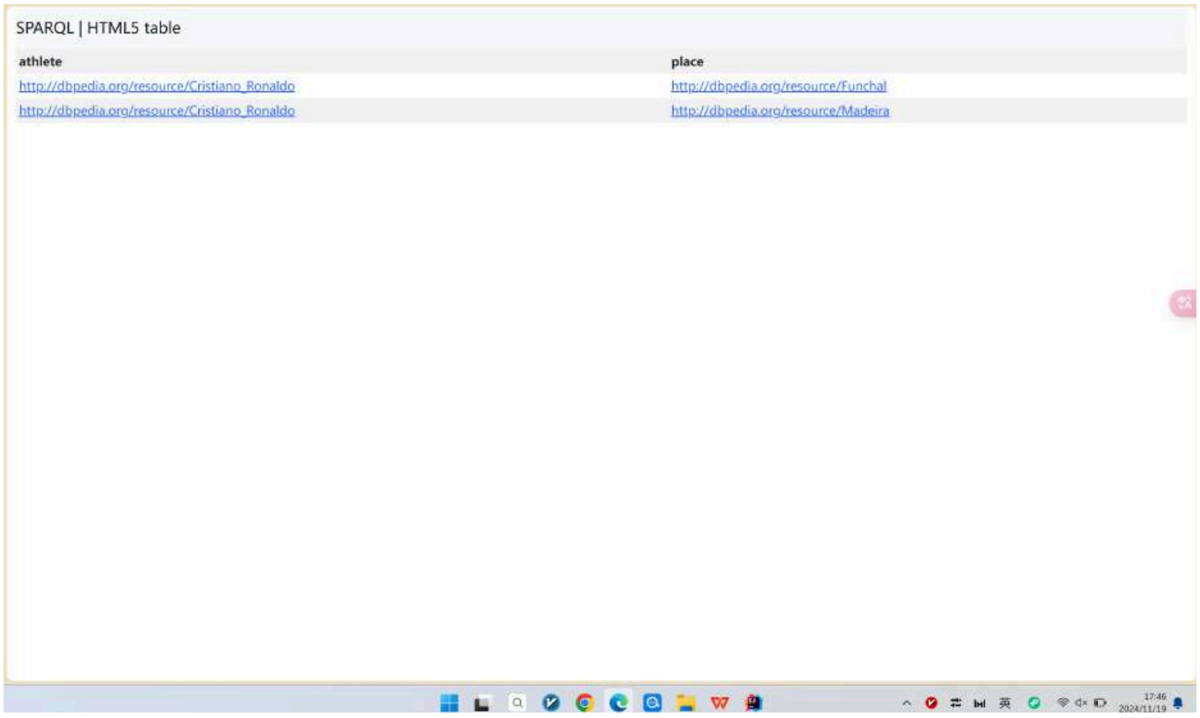
Task 1



SPARQL | HTML5 table

athlete
http://dbpedia.org/resource/Cristiano_Ronaldo
http://dbpedia.org/resource/Category:Cristiano_Ronaldo

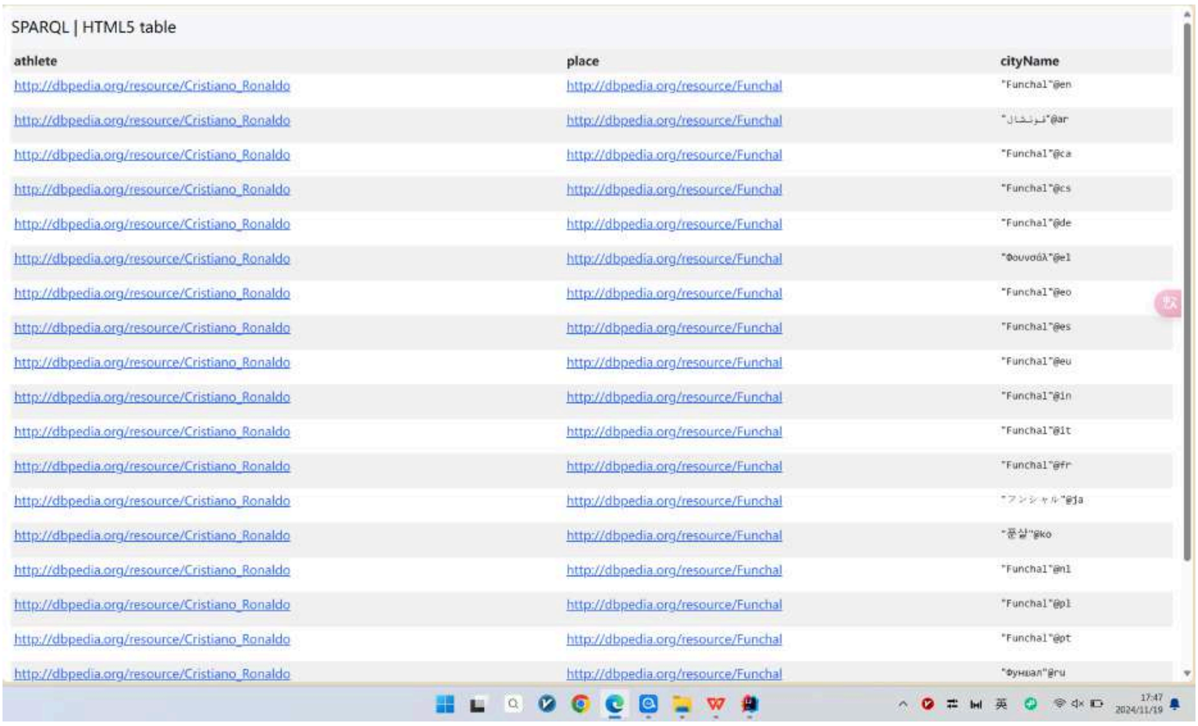
17:46
2024/11/19



SPARQL | HTML5 table

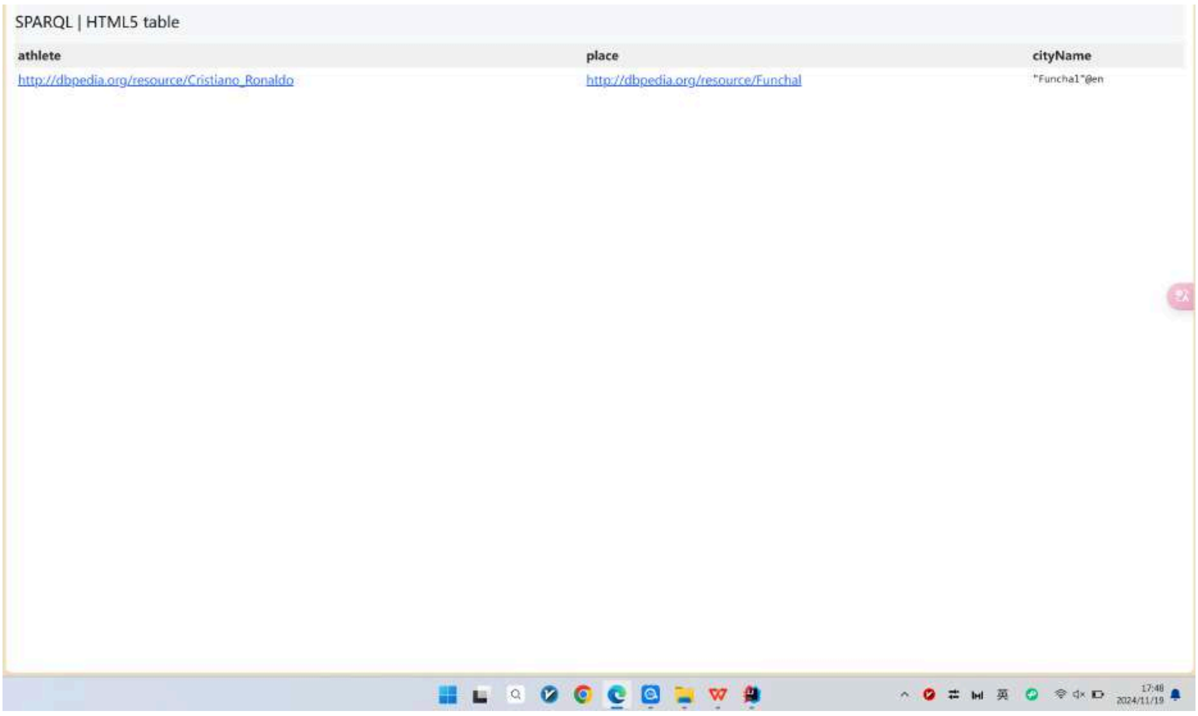
athlete	place
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Madeira

17:46
2024/11/19



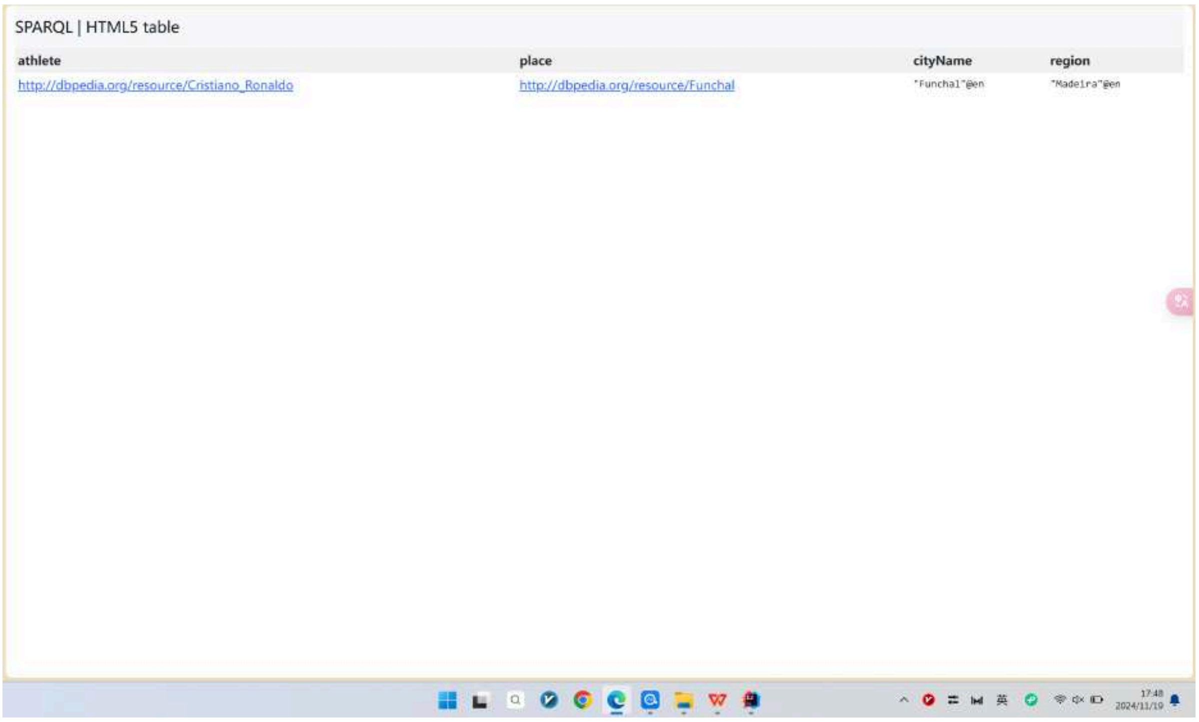
SPARQL | HTML5 table

athlete	place	cityName
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"Funchal"@en
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"فونشال"@ar
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"Funchal"@ca
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"Funchal"@cs
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"Funchal"@de
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"Φουνσάλ"@el
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"Funchal"@eo
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"Funchal"@es
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"Funchal"@eu
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"Funchal"@in
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"Funchal"@it
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"Funchal"@fr
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"フンシャル"@ja
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"푼샬"@ko
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"Funchal"@nl
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"Funchal"@pl
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"Funchal"@pt
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"Фуншал"@ru



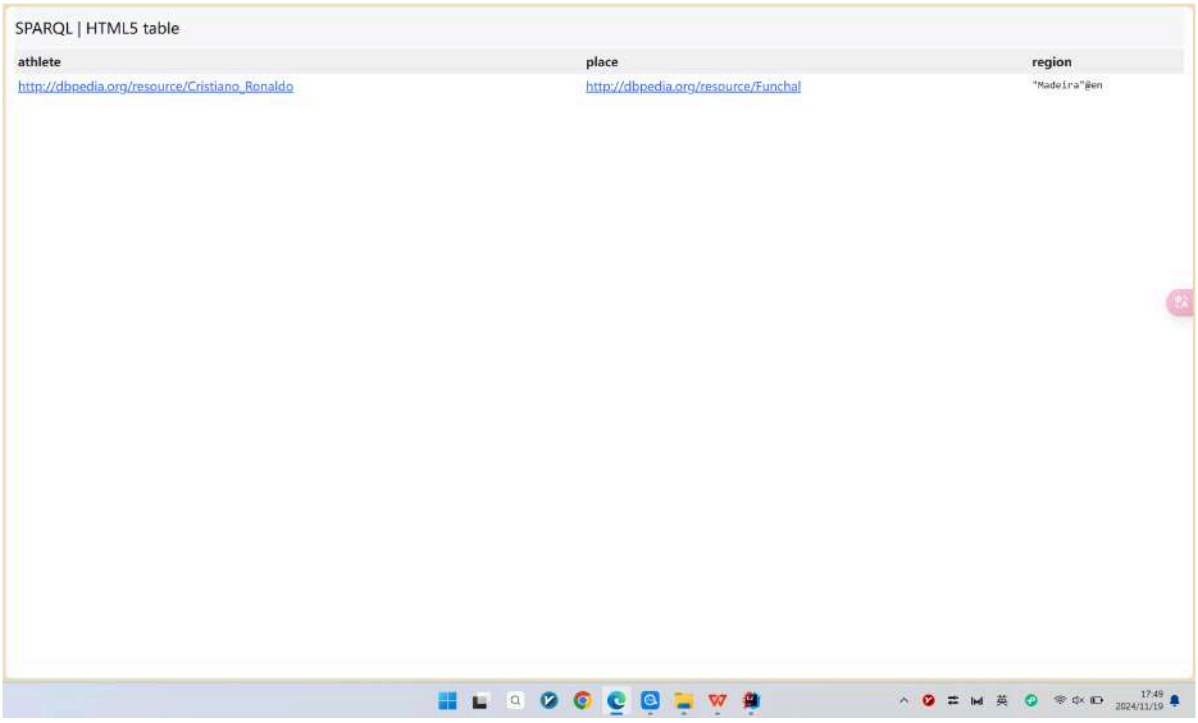
SPARQL | HTML5 table

athlete	place	cityName
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"Funchal"@en



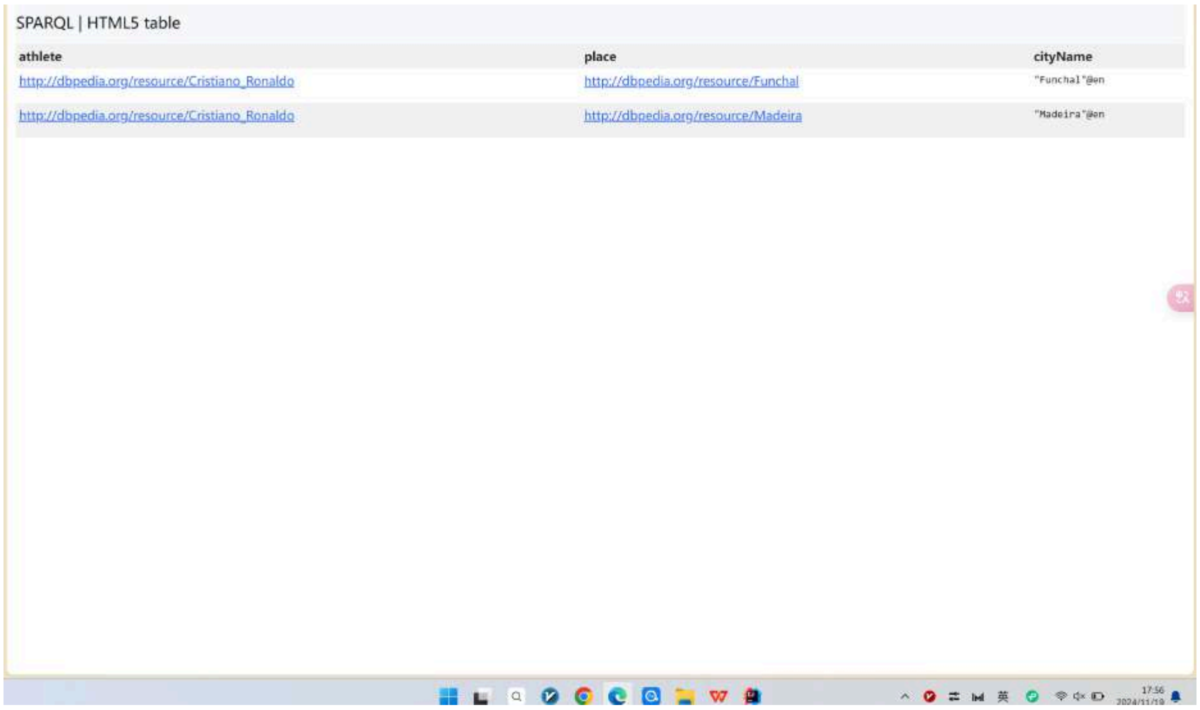
SPARQL | HTML5 table

athlete	place	cityName	region
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"Funchal"@en	"Madeira"@en



SPARQL | HTML5 table

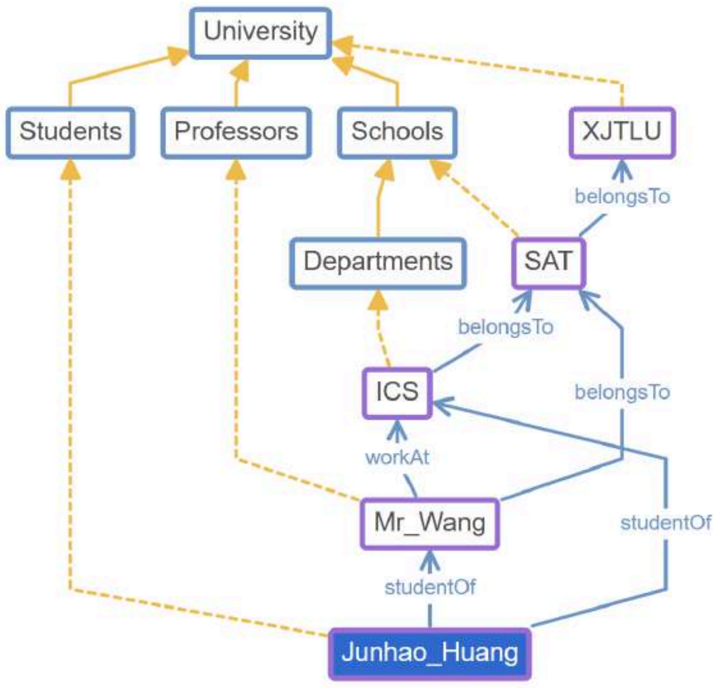
athlete	place	region
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"Madeira"@en



SPARQL | HTML5 table

athlete	place	cityName
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Funchal	"Funchal"@en
http://dbpedia.org/resource/Cristiano_Ronaldo	http://dbpedia.org/resource/Madeira	"Madeira"@en

Task 2



The RDF code of the graph:

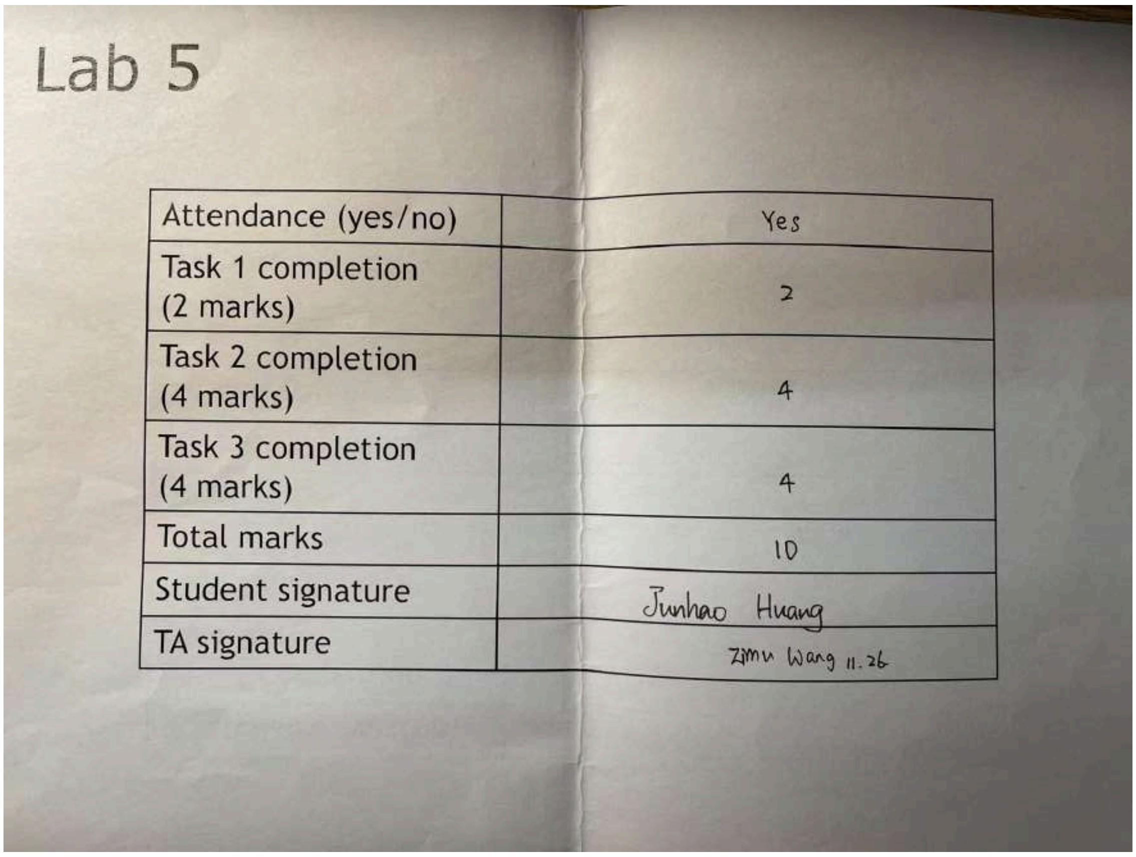
```
<?xml version="1.0"?>
<rdf:RDF xmlns="urn:webprotege:ontology:0043199f-c609-4775-9f0a-efbb33303640#"
     xml:base="urn:webprotege:ontology:0043199f-c609-4775-9f0a-efbb33303640"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     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/RBKKlJzgB7nwMXJr8uXqvII -->

    <owl:ObjectProperty
rdf:about="http://webprotege.stanford.edu/RBKKlJzgB7nwMXJr8uXqvII">
        <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/R9S46fKTuMhMBwYOWLnROQA -->

    <owl:Class
rdf:about="http://webprotege.stanford.edu/R9S46fKTuMhMBwYOWLnROQA">
        <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/RDNNFK313i7wkhoJFKj3rFR -->

    <owl:Class
rdf:about="http://webprotege.stanford.edu/RDNNFK313i7wkhoJFKj3rFR">
        <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/RrhWazT2CcY7ClJ611CHQh -->

    <owl:Class rdf:about="http://webprotege.stanford.edu/RrhWazT2CcY7ClJ611CHQh">
        <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/RDNNFK313i7wkhoJFKj3rFR"/>
        <webprotege:RBKKlJzgB7nwMXJr8uXqvII
rdf:resource="http://webprotege.stanford.edu/R9a1823UiUZOOBjfikJJHPE"/>
        <rdfs:label>ICS</rdfs:label>
    </owl:NamedIndividual>



    <!-- http://webprotege.stanford.edu/R9a1823UiUZOOBjfikJJHPE -->

    <owl:NamedIndividual
rdf:about="http://webprotege.stanford.edu/R9a1823UiUZOOBjfikJJHPE">
        <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/RrhWazT2CcY7ClJ611CHQh"/>
        <webprotege:R9PoKwuK24PwAYyqbtMFIXu
rdf:resource="http://webprotege.stanford.edu/R88U3jjQfVkgofoVEef261U"/>
        <webprotege:RBKKlJzgB7nwMXJr8uXqvII
rdf:resource="http://webprotege.stanford.edu/R9a1823UiUZOOBjfikJJHPE"/>
```

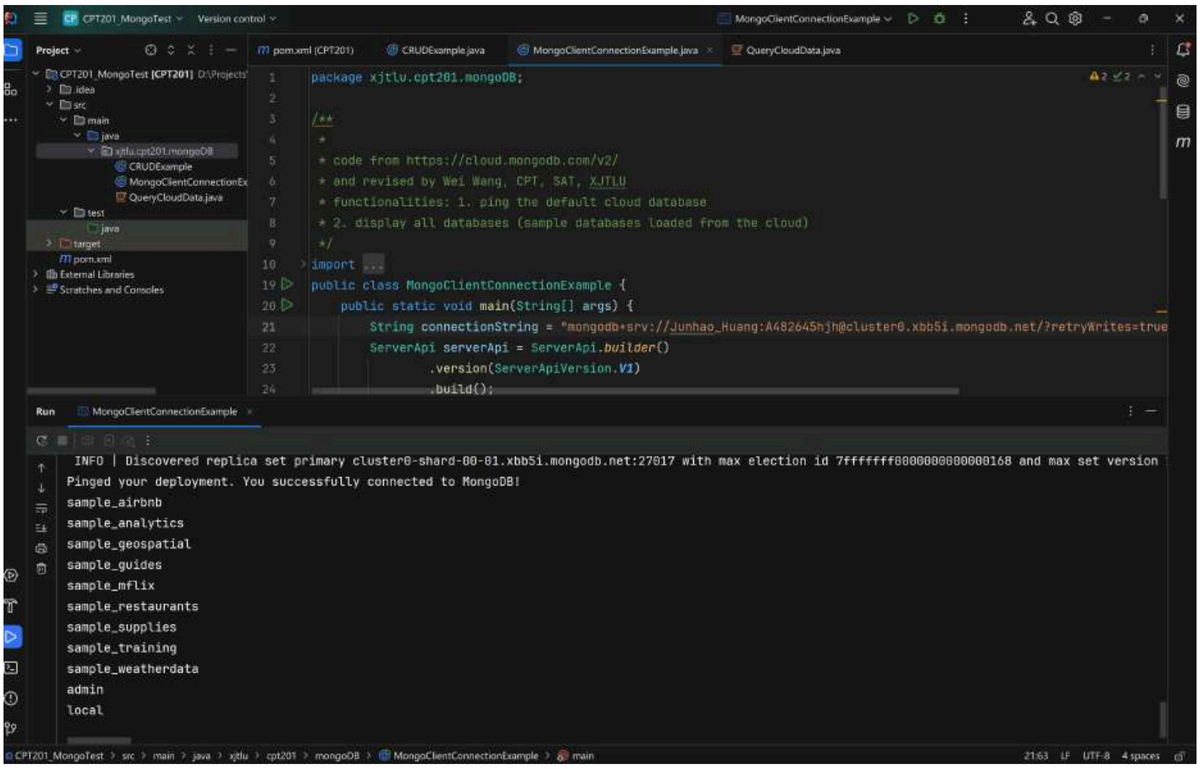
```
        <rdfs:label>Mr_Wang</rdfs:label>
    </owl:NamedIndividual>



    <!-- http://webprotege.stanford.edu/RDdkcUe6hWpzcyWJCXgufjX -->

    <owl:NamedIndividual
rdf:about="http://webprotege.stanford.edu/RDdkcUe6hWpzcyWJCXgufjX">
        <rdf:type
rdf:resource="http://webprotege.stanford.edu/RCbv9PMZpvT7R0q3nuG32iy"/>
        <rdfs:label>XJTLU</rdfs:label>
    </owl:NamedIndividual>



    <!-- http://webprotege.stanford.edu/REGNMENmTcCbocir7TVyL6 -->

    <owl:NamedIndividual
rdf:about="http://webprotege.stanford.edu/REGNMENmTcCbocir7TVyL6">
        <rdf:type
rdf:resource="http://webprotege.stanford.edu/R9S46fKTuMhMBwY0WLnROQA"/>
        <webprotege:RCL5CAkBFBKX0CW88JcRXHH
rdf:resource="http://webprotege.stanford.edu/R88U3jjQfVkgofoVEef261U"/>
        <webprotege:RCL5CAkBFBKX0CW88JcRXHH
rdf:resource="http://webprotege.stanford.edu/RBBDvwUUedRODwdOMy43uz4"/>
        <rdfs:label>Junhao_Huang</rdfs:label>
    </owl:NamedIndividual>
</rdf:RDF>



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

Lab 5 Java and MongoDB



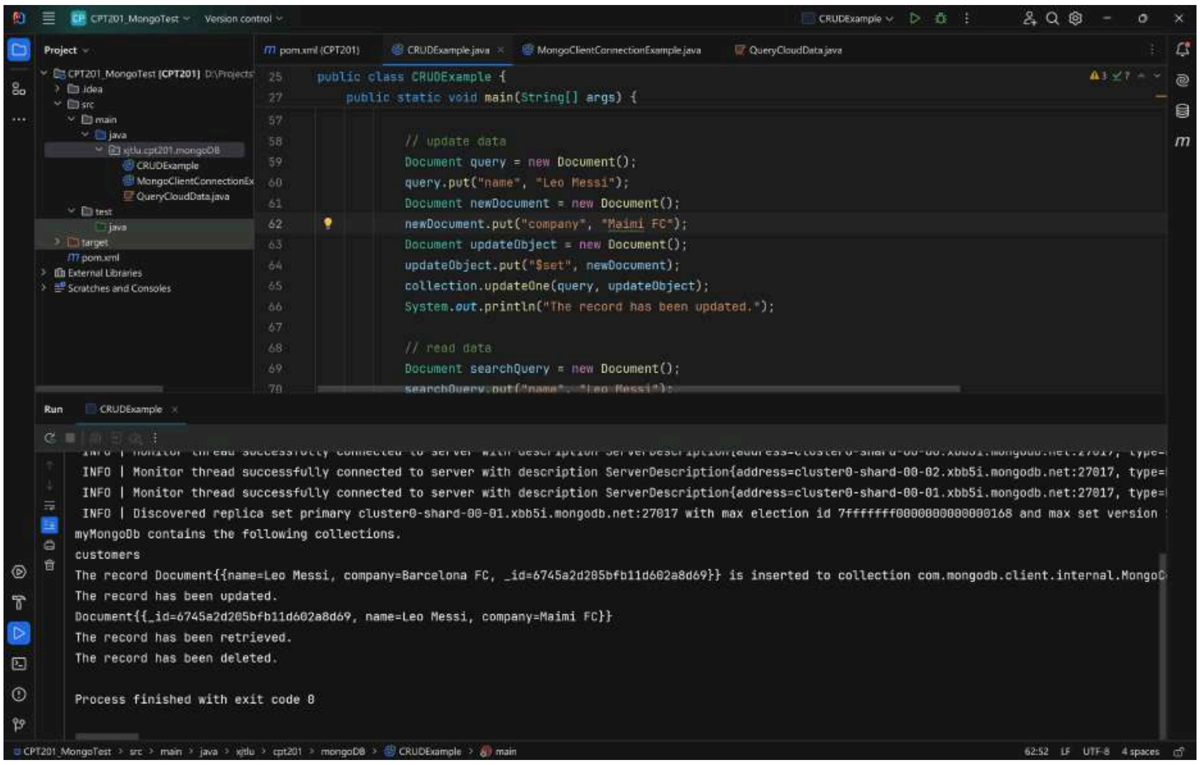
Lab 5

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	Zimu Wang 11.26



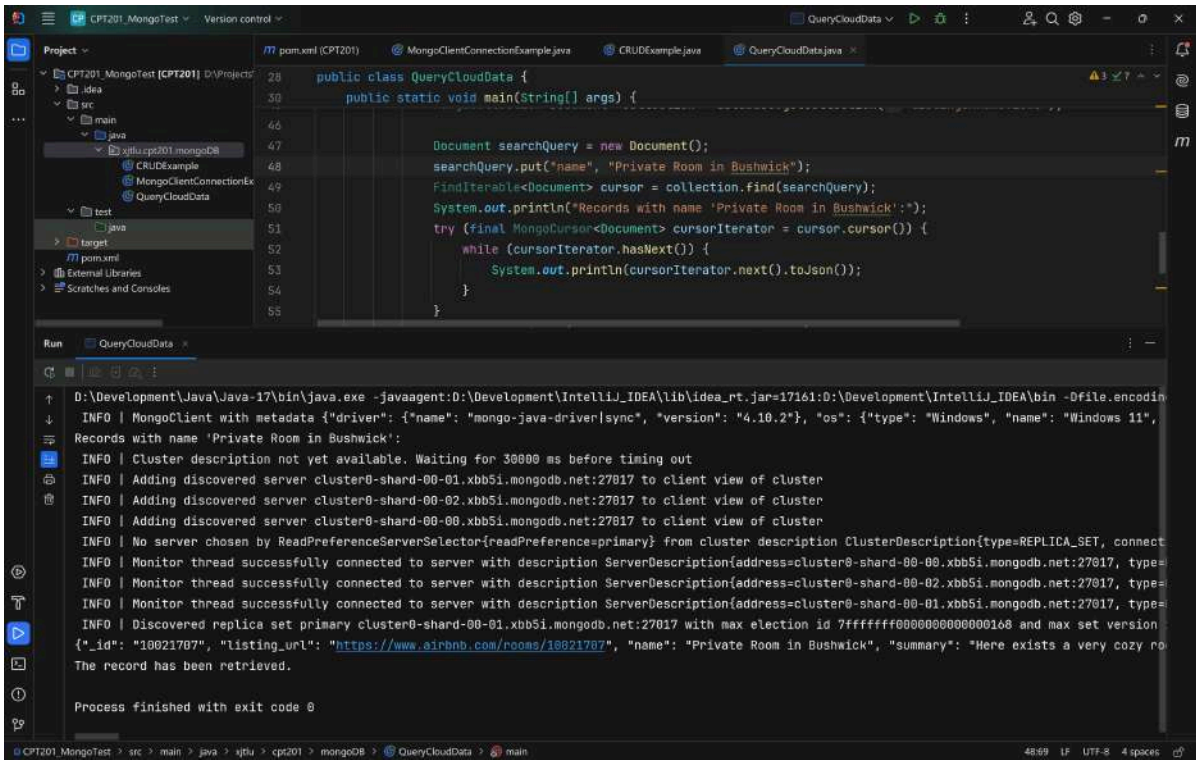
```java
package xjtlu.cpt201.mongoDB;

/**
 *
 * code from https://cloud.mongodb.com/v2/
 * and revised by Wei Wang, CPT, SAT, XJTLU
 * functionalities: 1. ping the default cloud database
 * 2. display all databases (sample databases loaded from the cloud)
 */
import ...
public class MongoClientConnectionExample {
    public static void main(String[] args) {
        String connectionString = "mongodb+srv://Junhao_Huang:A482645hjh@cluster0.xbb5i.mongodb.net/?retryWrites=true
        ServerApi serverApi = ServerApi.builder()
                .version(ServerApiVersion.V1)
                .build();
```

```
 INFO | Discovered replica set primary cluster0-shard-00-01.xbb5i.mongodb.net:27017 with max election id 7fffffff0000000000000168 and max set version
Pinged your deployment. You successfully connected to MongoDB!
sample_airbnb
sample_analytics
sample_geospatial
sample_guides
sample_mflix
sample_restaurants
sample_supplies
sample_training
sample_weatherdata
admin
local
```



```java
public class CRUDExample {
    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");
```

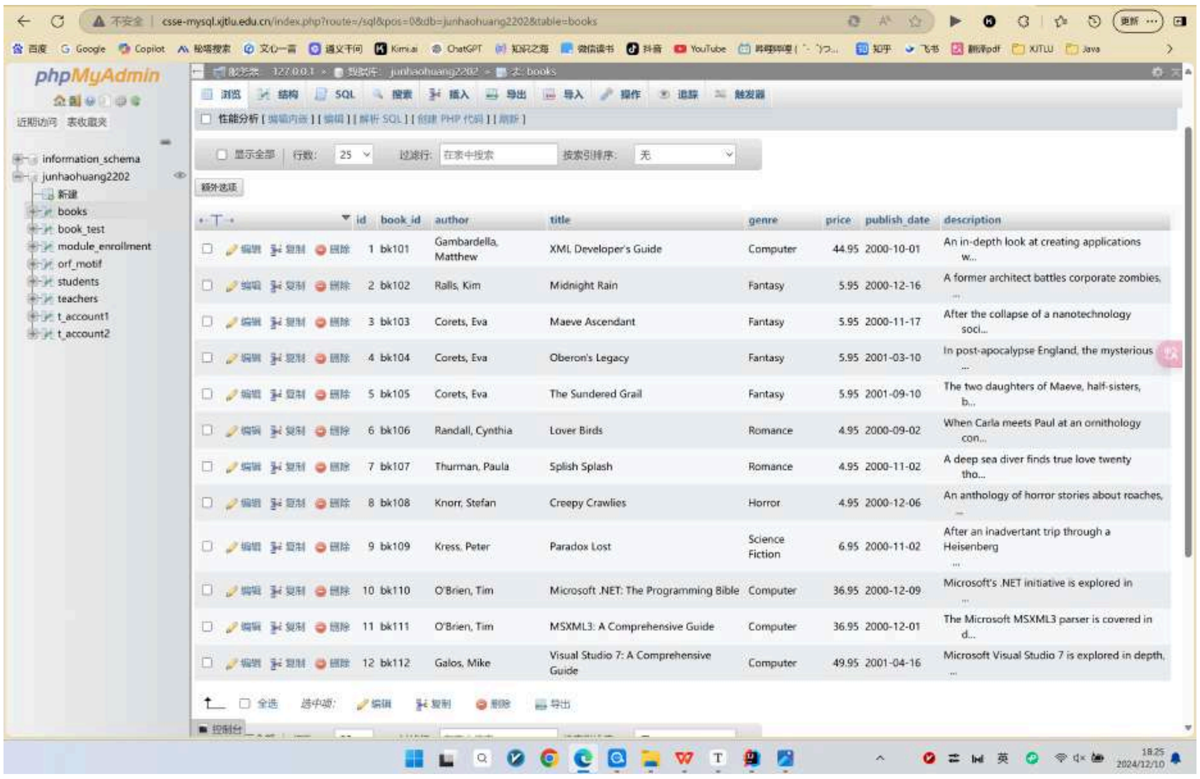
```
 INFO | Monitor thread successfully connected to server with description ServerDescription{address=cluster0-shard-00-00.xbb5i.mongodb.net:27017, type=
 INFO | Monitor thread successfully connected to server with description ServerDescription{address=cluster0-shard-00-02.xbb5i.mongodb.net:27017, type=
 INFO | Monitor thread successfully connected to server with description ServerDescription{address=cluster0-shard-00-01.xbb5i.mongodb.net:27017, type=
 INFO | Discovered replica set primary cluster0-shard-00-01.xbb5i.mongodb.net:27017 with max election id 7fffffff0000000000000168 and max set version
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.MongoC
The record has been updated.
Document{{_id=6745a2d205bfb11d602a8d69, name=Leo Messi, company=Maimi FC}}
The record has been retrieved.
The record has been deleted.

Process finished with exit code 0
```



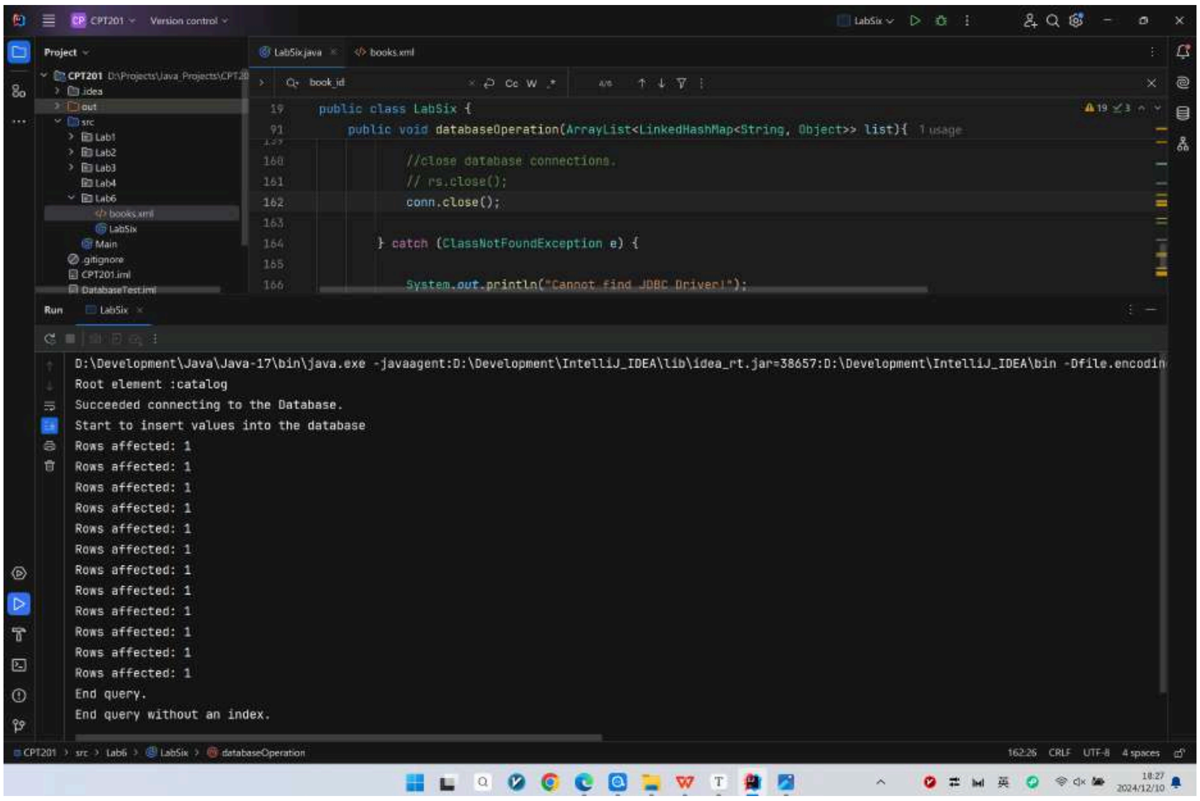
```java
public class QueryCloudData {
    public static void main(String[] args) {

                Document searchQuery = new Document();
                searchQuery.put("name", "Private Room in Bushwick");
                FindIterable<Document> cursor = collection.find(searchQuery);
                System.out.println("Records with name 'Private Room in Bushwick':");
                try (final MongoCursor<Document> cursorIterator = cursor.cursor()) {
                    while (cursorIterator.hasNext()) {
                        System.out.println(cursorIterator.next().toJson());
                    }
                }
```

```
D:\Development\Java\Java-17\bin\java.exe -javaagent:D:\Development\IntelliJ_IDEA\lib\idea_rt.jar=17161:D:\Development\IntelliJ_IDEA\bin -Dfile.encodin
 INFO | MongoClient with metadata {"driver": {"name": "mongo-java-driver|sync", "version": "4.10.2"}, "os": {"type": "Windows", "name": "Windows 11",
Records with name 'Private Room in Bushwick':
 INFO | Cluster description not yet available. Waiting for 30000 ms before timing out
 INFO | Adding discovered server cluster0-shard-00-01.xbb5i.mongodb.net:27017 to client view of cluster
 INFO | Adding discovered server cluster0-shard-00-02.xbb5i.mongodb.net:27017 to client view of cluster
 INFO | Adding discovered server cluster0-shard-00-00.xbb5i.mongodb.net:27017 to client view of cluster
 INFO | No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=REPLICA_SET, connect
 INFO | Monitor thread successfully connected to server with description ServerDescription{address=cluster0-shard-00-00.xbb5i.mongodb.net:27017, type=
 INFO | Monitor thread successfully connected to server with description ServerDescription{address=cluster0-shard-00-02.xbb5i.mongodb.net:27017, type=
 INFO | Monitor thread successfully connected to server with description ServerDescription{address=cluster0-shard-00-01.xbb5i.mongodb.net:27017, type=
 INFO | Discovered replica set primary cluster0-shard-00-01.xbb5i.mongodb.net:27017 with max election id 7fffffff0000000000000168 and max set version
{"_id": "10021707", "listing_url": "https://www.airbnb.com/rooms/10021707", "name": "Private Room in Bushwick", "summary": "Here exists a very cozy ro
The record has been retrieved.

Process finished with exit code 0
```

Lab 6 XML and MongoDB



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 inadvertant 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, ...



```java
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!");
```

```
D:\Development\Java\Java-17\bin\java.exe -javaagent:D:\Development\IntelliJ_IDEA\lib\idea_rt.jar=38657:D:\Development\IntelliJ_IDEA\bin -Dfile.encodin
Root element :catalog
Succeeded connecting to the Database.
Start to insert values into the database
Rows affected: 1
Rows affected: 1
Rows affected: 1
Rows affected: 1
Rows affected: 1
Rows affected: 1
Rows affected: 1
Rows affected: 1
Rows affected: 1
Rows affected: 1
Rows affected: 1
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

```java
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);
                   // 讲当前的 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"));
        }
        labSix.databaseOperation(result);
    }
}
```