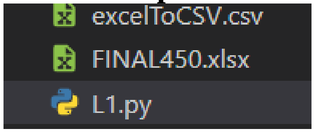
BDA-LAB01

-Jainil Trivedi (CE166)

Tasks

1. **Given the spreadsheet file convert it into a csv**



excelToCSV.csv

FINAL450.xlsx

L1.py

Code:

```
import pandas as pd

excel = pd.read_excel('FINAL450.xlsx')
df = pd.DataFrame(excel)
df.to_csv("excelToCSV.csv",index=False)
```

2. **Import a csv into MySQL database table**

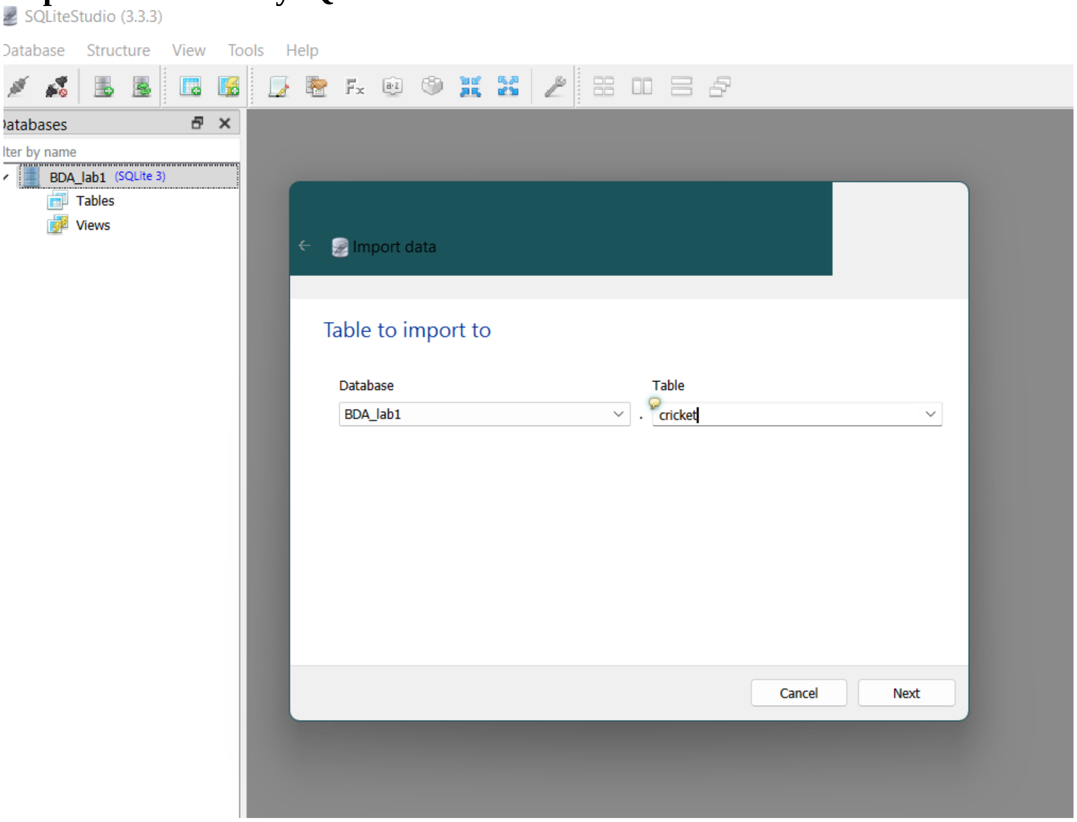
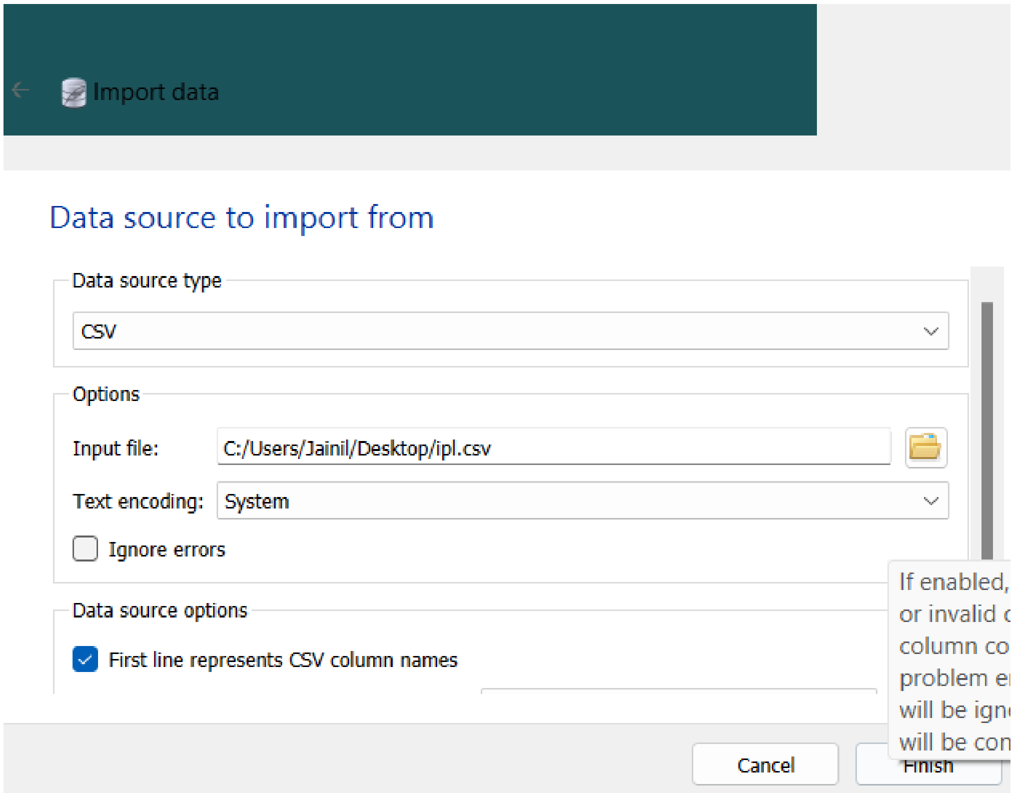


Table to import to

Database: BDA_lab1

Table: cricket



Import data

Data source to import from

Data source type

CSV

Options

Input file: C:/Users/Jainil/Desktop/ipl.csv

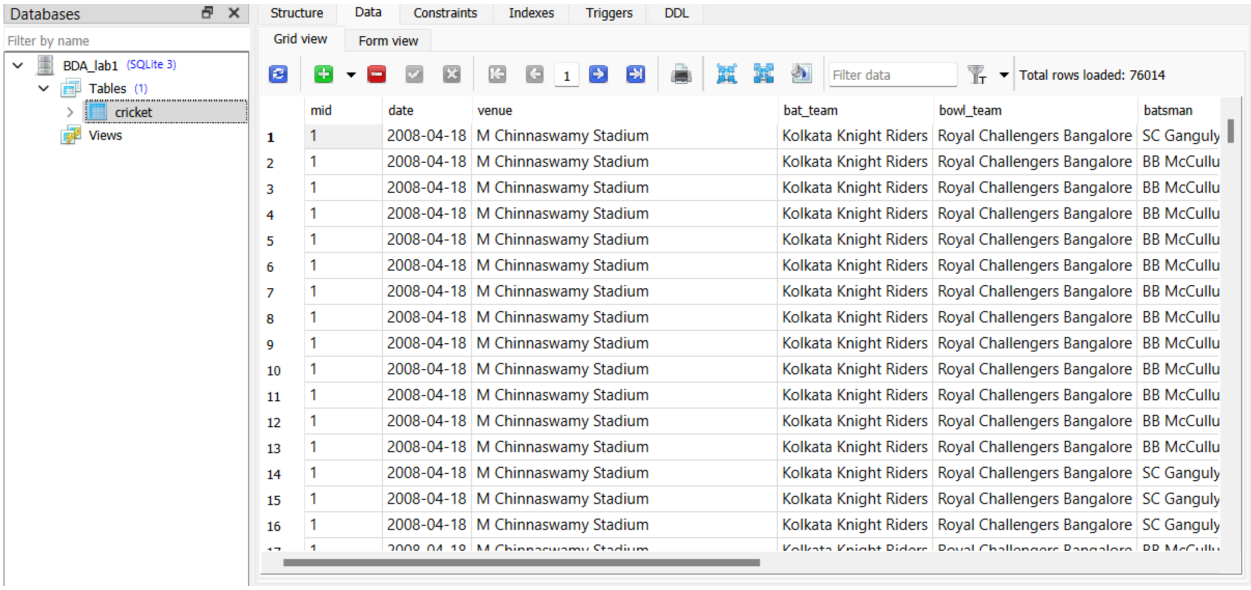
Text encoding: System

□ Ignore errors

Data source options

☑ First line represents CSV column names

If enabled,
or invalid c
column co
problem e
will be ign
will be con

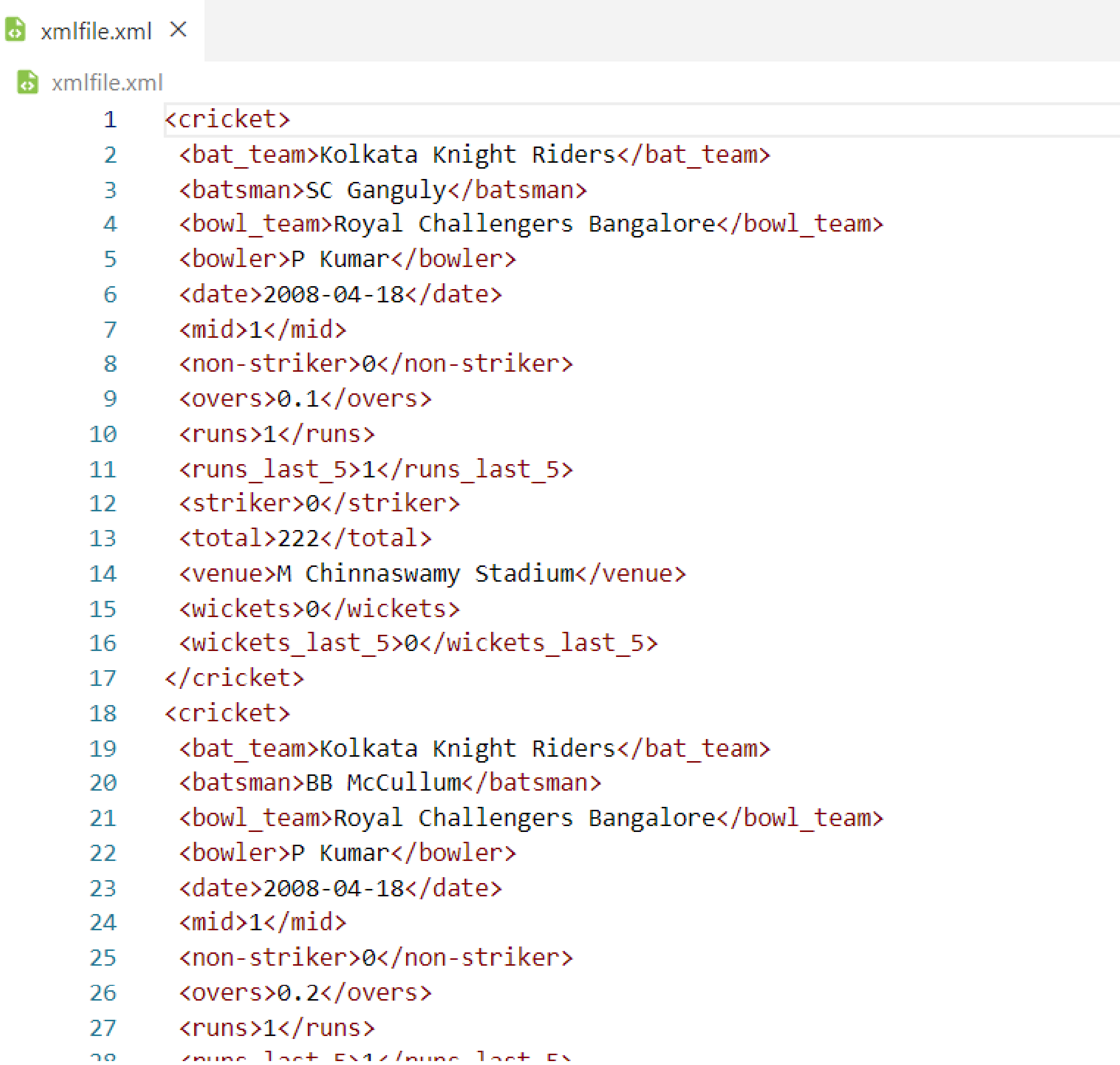


Total rows loaded: 76014

	mid	date	venue	bat_team	bowl_team	batsman
1	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	SC Ganguly
2	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullu
3	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullu
4	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullu
5	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullu
6	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullu
7	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullu
8	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullu
9	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullu
10	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullu
11	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullu
12	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullu
13	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullu
14	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	SC Ganguly
15	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	SC Ganguly
16	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	SC Ganguly
17	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullu

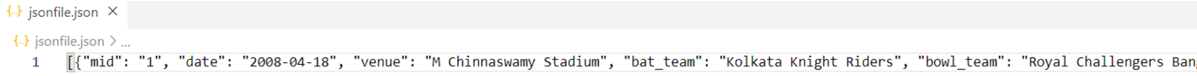
3. **Write a computer program to read records from database and generate data file.**

=> XML



```
<cricket>
 <bat_team>Kolkata Knight Riders</bat_team>
 <batsman>SC Ganguly</batsman>
 <bowl_team>Royal Challengers Bangalore</bowl_team>
 <bowler>P Kumar</bowler>
 <date>2008-04-18</date>
 <mid>1</mid>
 <non-striker>0</non-striker>
 <overs>0.1</overs>
 <runs>1</runs>
 <runs_last_5>1</runs_last_5>
 <striker>0</striker>
 <total>222</total>
 <venue>M Chinnaswamy Stadium</venue>
 <wickets>0</wickets>
 <wickets_last_5>0</wickets_last_5>
</cricket>
<cricket>
 <bat_team>Kolkata Knight Riders</bat_team>
 <batsman>BB McCullum</batsman>
 <bowl_team>Royal Challengers Bangalore</bowl_team>
 <bowler>P Kumar</bowler>
 <date>2008-04-18</date>
 <mid>1</mid>
 <non-striker>0</non-striker>
 <overs>0.2</overs>
 <runs>1</runs>
 <runs_last_5>1</runs_last_5>
```

⇨ **JSON**

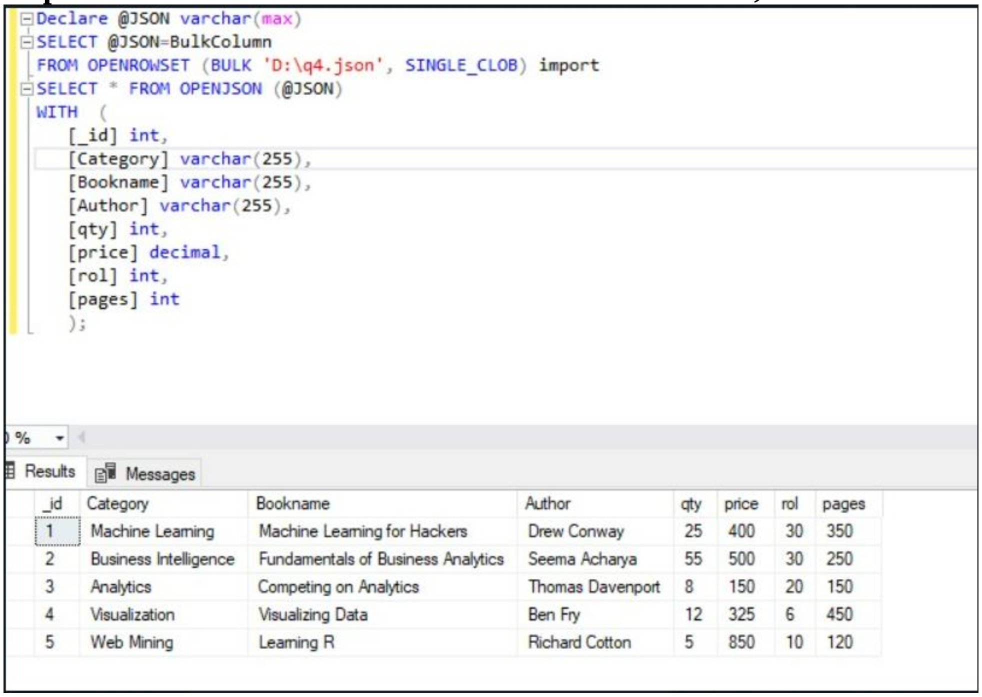


```
[{"mid": "1", "date": "2008-04-18", "venue": "M Chinnaswamy Stadium", "bat_team": "Kolkata Knight Riders", "bowl_team": "Royal Challengers Ban
```

```
import sqlite3
import json
from textwrap import wrap
from dict2xml import dict2xml as xmlify
conn =
sqlite3.connect("C:\\Users\\Jainil\\Downloads\\SQLiteStudio\\BDA_
lab1")
```

```
cur = conn.cursor()
cur.execute('SELECT * FROM cricket')

columns = list(map(lambda x:x[0],cur.description))
records = cur.fetchall()
record_list = []
for record in range(0,len(records)):
    record_dict = {}
    for column in range(0,len(columns)):
        record_dict[columns[column]] = records[record][column]
    record_list.append(record_dict)

json_obj = json.dumps(record_list)
xml_boj = xmlify(record_list,wrap='cricket',indent=" ")

json_file = open("jsonfile.json","w")
json_file.write(json_obj)

xml_file = open("xmlfile.xml","w")
xml_file.write(xml_boj)

conn.close()
```

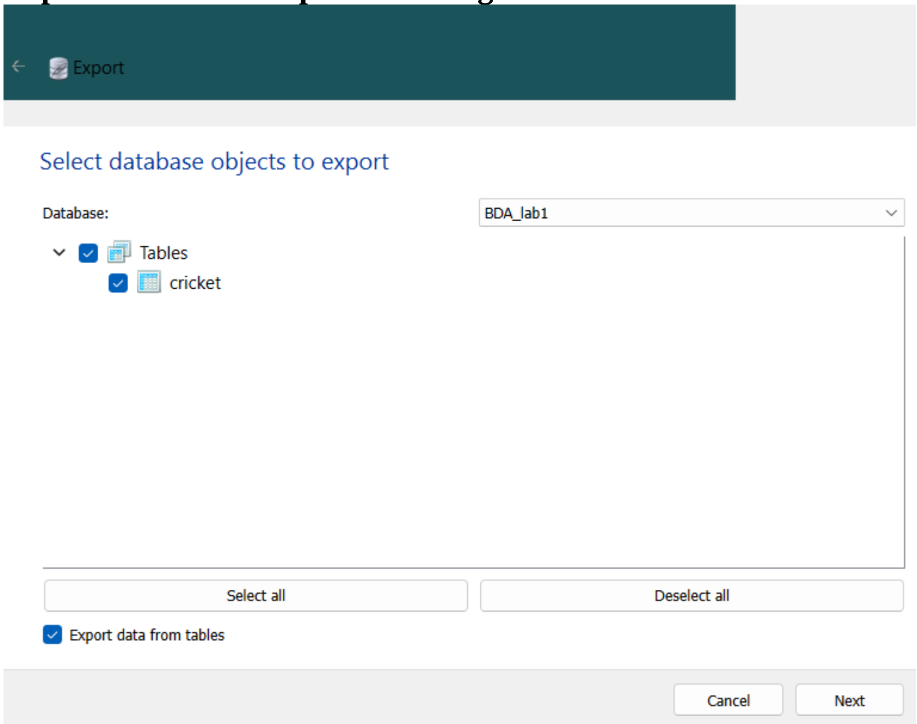
4. Import XML/JSON file into another database/table. i.e, MS Access. Oracle, etc.



```
Declare @JSON varchar(max)
SELECT @JSON=BulkColumn
FROM OPENROWSET (BULK 'D:\q4.json', SINGLE_CLOB) import
SELECT * FROM OPENJSON (@JSON)
WITH  (
    [_id] int,
    [Category] varchar(255),
    [Bookname] varchar(255),
    [Author] varchar(255),
    [qty] int,
    [price] decimal,
    [rol] int,
    [pages] int
    );
```

_id	Category	Bookname	Author	qty	price	rol	pages
1	Machine Learning	Machine Learning for Hackers	Drew Conway	25	400	30	350
2	Business Intelligence	Fundamentals of Business Analytics	Seema Acharya	55	500	30	250
3	Analytics	Competing on Analytics	Thomas Davenport	8	150	20	150
4	Visualization	Visualizing Data	Ben Fry	12	325	6	450
5	Web Mining	Learning R	Richard Cotton	5	850	10	120

5. Export database dump for data migration/archival

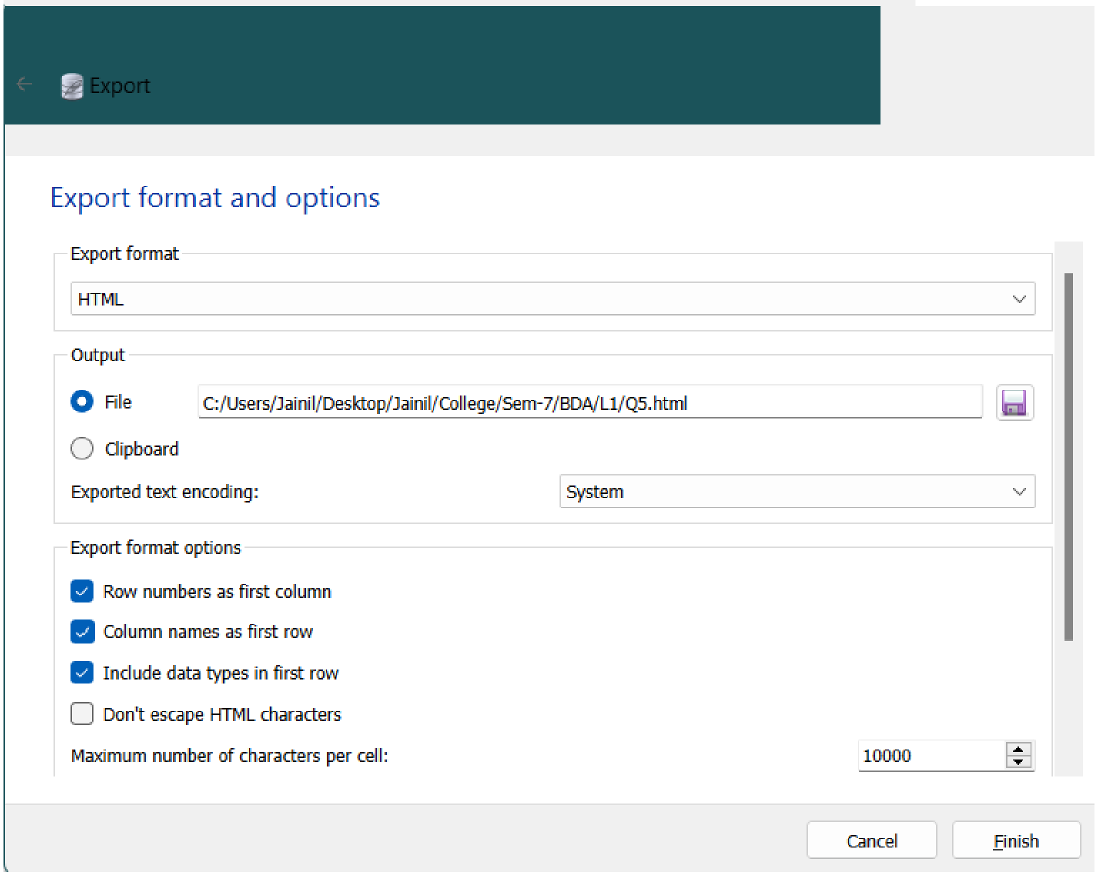


Select database objects to export

Database: BDA_lab1

- ☑ Tables
  - ☑ cricket

☑ Export data from tables



Export format and options

Export format: HTML

Output

● File C:/Users/Jainil/Desktop/Jainil/College/Sem-7/BDA/L1/Q5.html

○ Clipboard

Exported text encoding: System

Export format options

☑ Row numbers as first column

☑ Column names as first row

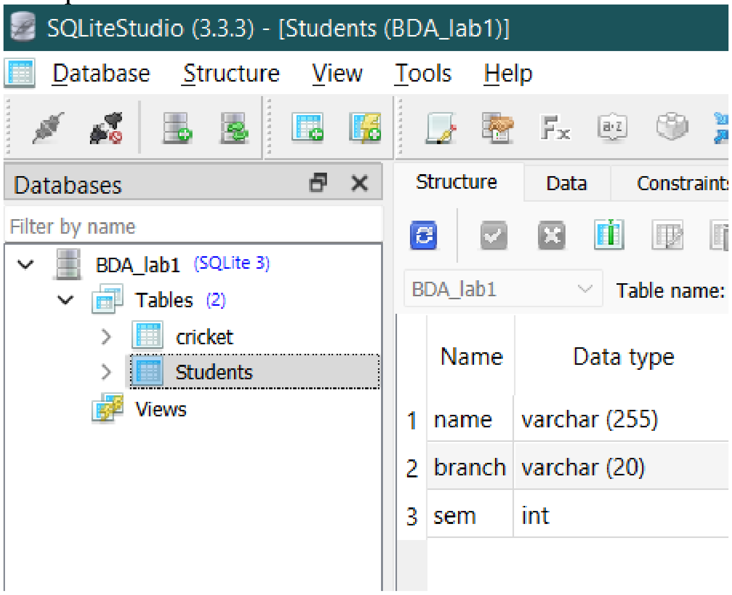
☑ Include data types in first row

□ Don't escape HTML characters

Maximum number of characters per cell: 10000

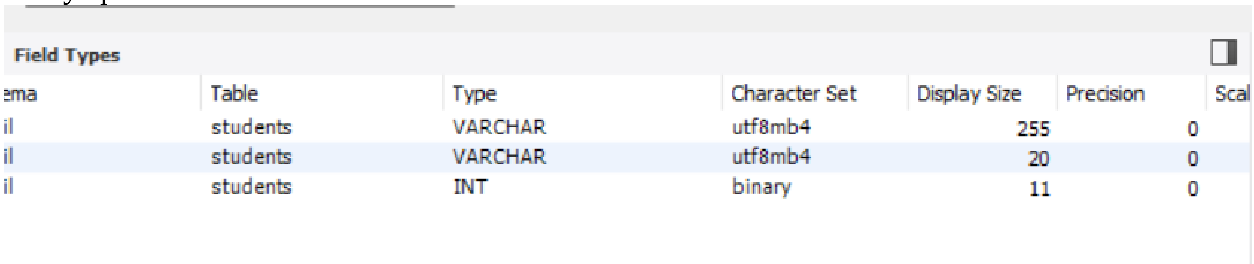
6. **Validate/Map data types across different database systems when migrating from one to another**

In SqliteStudio



	Name	Data type
1	name	varchar (255)
2	branch	varchar (20)
3	sem	int

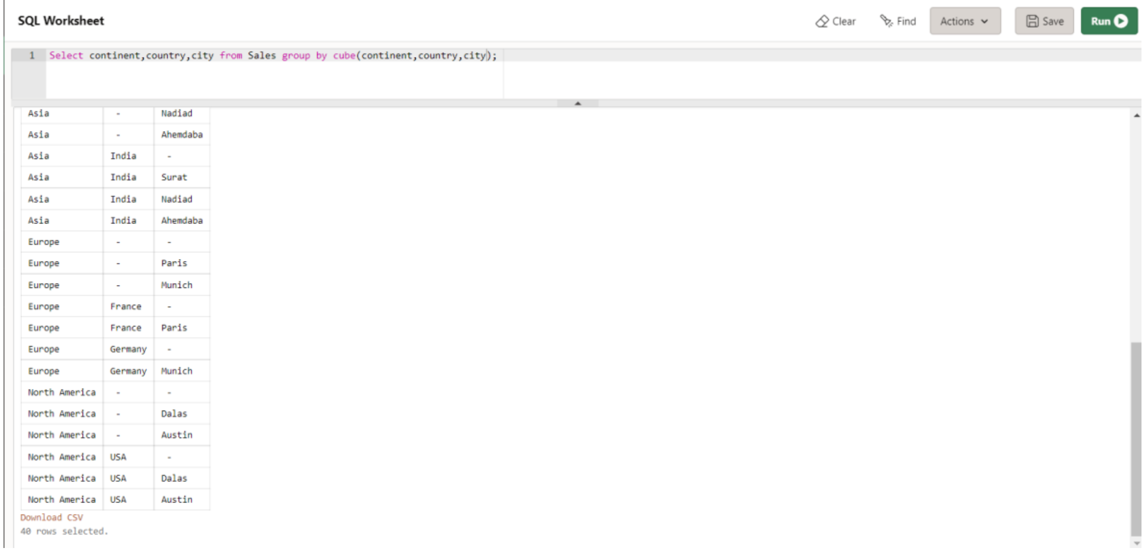
In mysql



ema	Table	Type	Character Set	Display Size	Precision	Scal
il	students	VARCHAR	utf8mb4	255	0	
il	students	VARCHAR	utf8mb4	20	0	
il	students	INT	binary	11	0	

7. **Represent Data Cube and perform operations. OLAP - Data Warehouse**

i) Cube



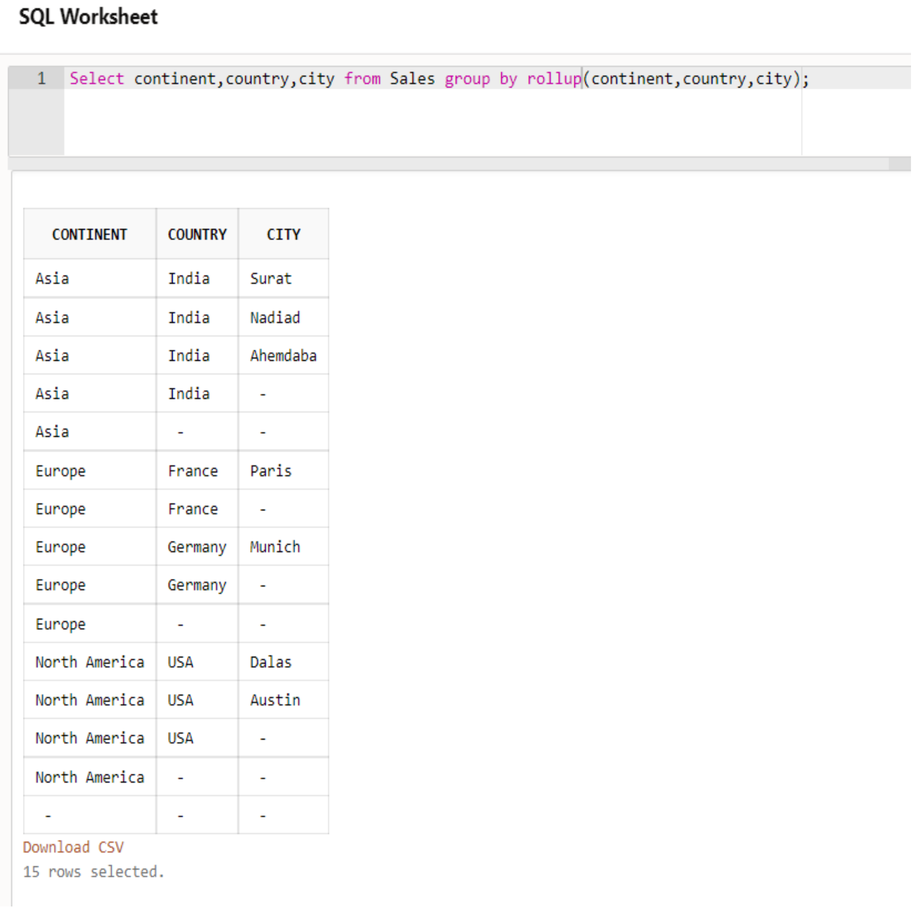
```
Select continent,country,city from Sales group by cube(continent,country,city);
```

Asia	-	Nadiad
Asia	-	Ahemdaba
Asia	India	-
Asia	India	Surat
Asia	India	Nadiad
Asia	India	Ahemdaba
Europe	-	-
Europe	-	Paris
Europe	-	Munich
Europe	France	-
Europe	France	Paris
Europe	Germany	-
Europe	Germany	Munich
North America	-	-
North America	-	Dalas
North America	-	Austin
North America	USA	-
North America	USA	Dalas
North America	USA	Austin

Download CSV

40 rows selected.

ii) Roll up



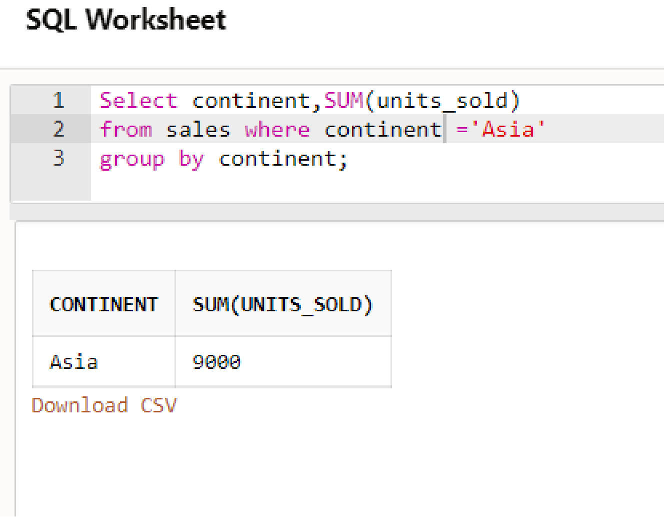
SQL Worksheet

```sql
Select continent,country,city from Sales group by rollup(continent,country,city);
```

CONTINENT	COUNTRY	CITY
Asia	India	Surat
Asia	India	Nadiad
Asia	India	Ahemdaba
Asia	India	-
Asia	-	-
Europe	France	Paris
Europe	France	-
Europe	Germany	Munich
Europe	Germany	-
Europe	-	-
North America	USA	Dalas
North America	USA	Austin
North America	USA	-
North America	-	-
-	-	-

```
15 rows selected.
```

iii) Slice

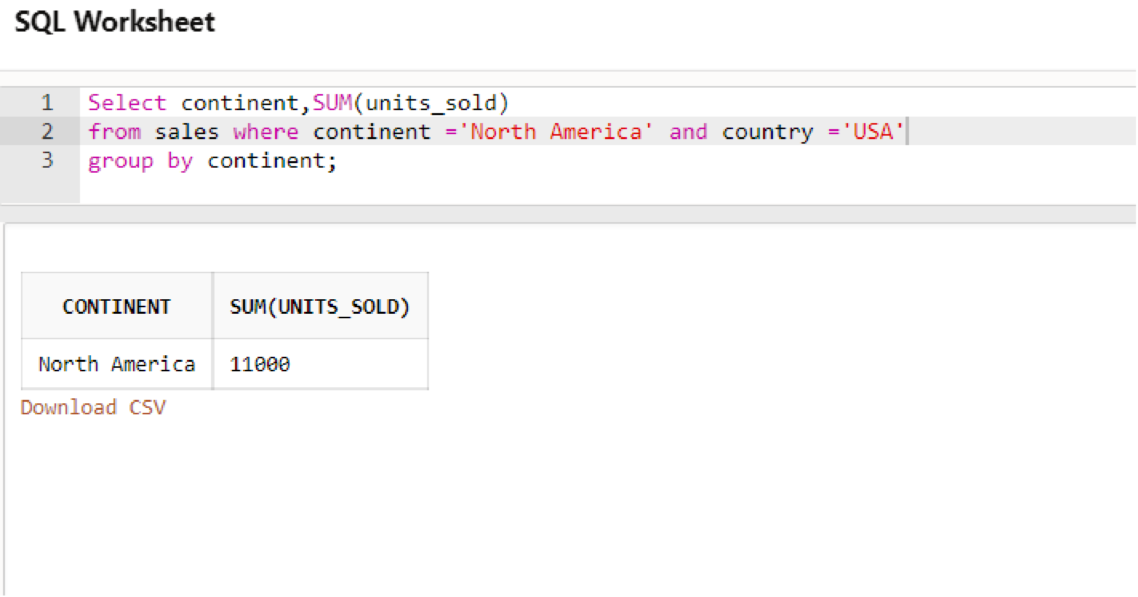


SQL Worksheet

```sql
Select continent,SUM(units_sold)
from sales where continent ='Asia'
group by continent;
```

CONTINENT	SUM(UNITS_SOLD)
Asia	9000

iv) Dice



SQL Worksheet

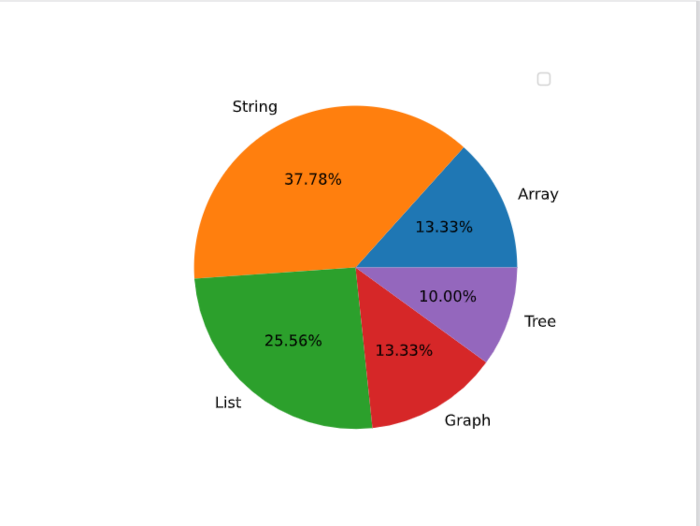
```
Select continent,SUM(units_sold)
from sales where continent ='North America' and country ='USA'
group by continent;
```

CONTINENT	SUM(UNITS_SOLD)
North America	11000

Download CSV

8. **Generate pdf report/ Use any visualization tool. i.e., pie chart, maps.**

```
import pandas as pd
import matplotlib.pyplot as plt

ds = pd.read_csv("Q8.csv")
topic = ds['Topic']
questions = ds['Questions']
x=list(topic)
y=list(questions)
plt.legend('topic','solved')
plt.pie(y,labels=x,autopct='%.2f%%')
plt.savefig("piechart_Q8.pdf",format="pdf")
```


String
Array
Tree
List
Graph