

1. Create a table attribute dataset and dress dataset
2. Do a bulk load for these two table for respective dataset
3. read these dataset in pandas as a dataframe
4. Convert attribute dataset in json format
5. Store this dataset into mongodb
6. in sql task try to perform left join operation with attribute dataset and dress dataset on column Dress_ID
7. Write a sql query to find out how many unique dress that we have based on dress id
8. Try to find out how many dress is having recommendation 0
9. Try to find out total dress sell for individual dress id
10. Try to find out a third highest most selling dress id

```
# 1. Create a table attribute dataset and dress dataset
```

```
-----
```

```
create table if not exists Attribute_Data (
```

```
Dress_ID int,  
Style varchar(30),  
Price varchar(30),  
Rating DECIMAL(2,1),  
Size varchar(30),  
Season varchar(30),  
NeckLine varchar(30),  
SleeveLength varchar(30),  
waixeline varchar(30),  
Material varchar(30),  
FabricType varchar(30),  
Decoration varchar(30),  
PatternType varchar(30),  
Recommendation int
```

```
)
```

```
create table if not exists Dress_Sales_Data (
```

```
Dress_ID bigint,  
`29-08-2013` int,  
`31-08-2013` int,  
`02-09-2013` int,  
`04-09-2013` int,  
`06-09-2013` int,  
`08-09-2013` int,  
`10-09-2013` int,  
`12-09-2013` int,  
`14-09-2013` int,  
`16-09-2013` int,  
`18-09-2013` int,  
`20-09-2013` int,
```

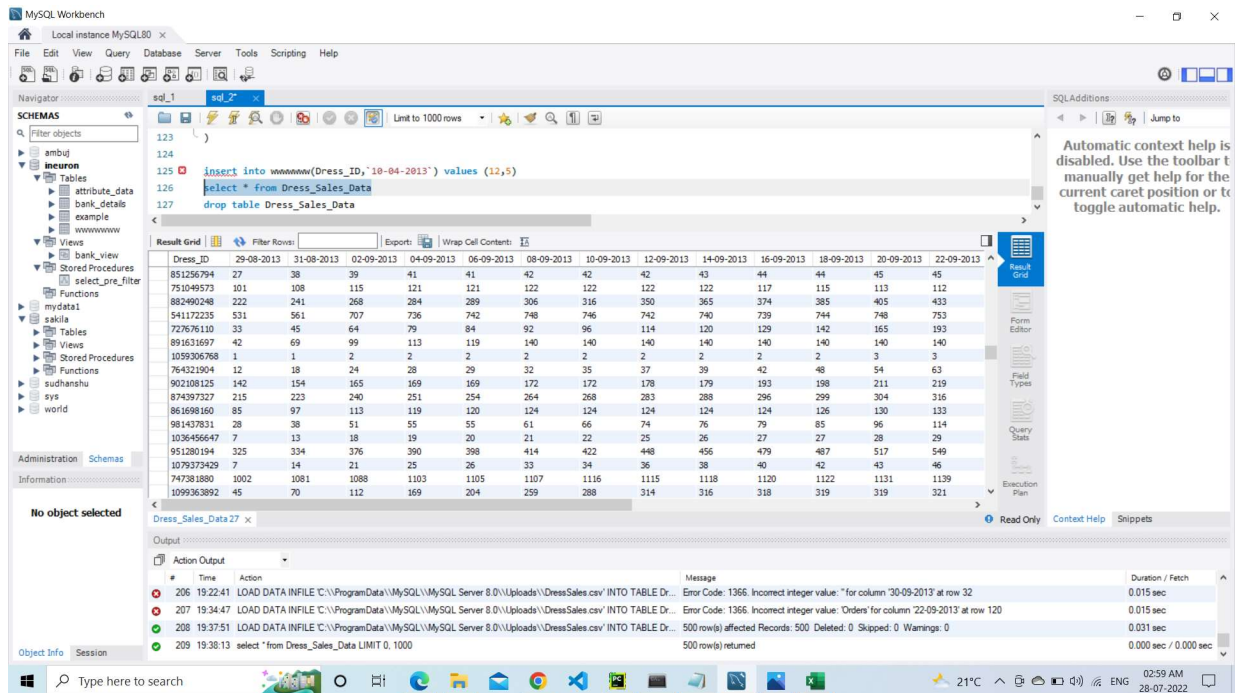
)

.....

```
LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server
8.0\\Uploads\\DressSales.csv' INTO TABLE Dress_Sales_Data
FIELDS TERMINATED BY ',' ENCLOSED BY '' ESCAPED BY '\\'
LINES TERMINATED BY '\\n'
IGNORE 1 lines;
```

The screenshot displays the MySQL Workbench environment. The top toolbar includes icons for file operations, editing, and database management. The left sidebar shows the 'SCHEMAS' tree with a filter for 'neuron'. The main workspace shows a table with 20 rows of data. The bottom status bar indicates '500 rows affected'.

Dress_ID	Style	Price	Rating	Size	Season	NeckLine	SleeveLength	waistline	Material	FabricType	Decoration	PatternType	Recommendation
1005032852	Sexy	Low	4.6	M	Summer	o-neck	sleeveless	empire	null	chiffon	ruffles	animal	1
1212192089	Casual	Low	0.0	L	Summer	o-neck	Petal	natural	microfiber	null	ruffles	animal	0
1190380701	Vintage	High	0.0	L	Autumn	o-neck	full	natural	polyester	null	null	print	0
966005983	Brief	Average	4.6	L	Spring	o-neck	full	natural	silk	chiffon	embroidary	print	1
876339541	cute	Low	4.5	M	Summer	o-neck	butterfly	natural	chiffonfabric	chiffon	bow	dot	0
1068323468	bohemian	Low	0.0	M	Summer	v-neck	sleeveless	empire	null	null	print	0	0
1220707172	Casual	Average	0.0	XL	Summer	o-neck	full	null	cotton	null	null	sold	0
1219677488	Novelty	Average	0.0	free	Autumn	o-neck	short	natural	polyester	broaddoth	lace	null	0
1113094204	Flare	Average	0.0	free	Spring	v-neck	short	empire	cotton	broaddoth	beading	sold	1
985292672	bohemian	Low	0.0	free	Summer	v-neck	sleeveless	natural	nylon	chiffon	null	null	1
1117297701	party	Average	5.0	free	Summer	o-neck	full	null	polyester	broaddoth	lace	sold	0
898481530	Flare	Average	0.0	free	Spring	v-neck	short	null	nylon	chiffon	null	animal	0
957723897	sexy	Low	4.7	M	Winter	o-neck	threequarter	null	null	chiffon	lace	print	1
749031896	vintage	Average	4.8	M	Summer	o-neck	short	empire	cotton	jersey	null	animal	1
1055411544	Casual	Low	5.0	M	Summer	boat-neck	full	null	cotton	null	sashes	sold	0
1162628131	Casual	Low	0.0	free	Winter	boat-neck	full	null	other	other	lace	sold	0
524514841	cute	Average	4.7	L	spring	o-neck	short	null	cotton	null	sashes	sold	1
820467746	bohemian	Medium	5.0	free	Autumn	o-neck	full	natural	null	hollowout	patchwork	1	0



3. read these dataset in pandas as a dataframe

In [33]:

```

import mysql.connector as connection
import pandas as pd

mydb = connection.connect(host="localhost", user="root", passwd="Mishra6392@")
print(mydb)
cursor = mydb.cursor()
#cursor.execute("show databases")
#print(cursor.fetchall())

```

<mysql.connector.connection_cext.CMySQLConnection object at 0x0000010A3A11D5E0>

```
In [32]: cursor.execute('SELECT * FROM ineuron.Attribute_Data')
table_rows=cursor.fetchall()
Attribute_df = pd.DataFrame(table_rows)
Attribute_df
```

```
Out[32]:
```

	0	1	2	3	4	5	6	7	8	9	.
0	1006032852	Sexy	Low	4.6	M	Summer	O-neck	sleeveless	empire	null	chiff
1	1212192089	Casual	Low	0.0	L	Summer	O-neck	Petal	natural	microfiber	n
2	1190380701	vintage	High	0.0	L	Autumn	O-neck	full	natural	polyester	n
3	966005983	Brief	Average	4.6	L	Spring	O-neck	full	natural	silk	chiff
4	876339541	cute	Low	4.5	M	Summer	O-neck	butterfly	natural	chiffonfabric	chiff
...
495	713391965	Casual	Low	4.7	M	Spring	O-neck	full	natural	polyester	n
496	722565148	Sexy	Low	4.3	free	Summer	O-neck	full	empire	cotton	n
497	532874347	Casual	Average	4.7	M	Summer	V-neck	full	empire	cotton	n
498	655464934	Casual	Average	4.6	L	winter	boat-neck	sleeveless	empire	silk	broadclo
499	919930954	Casual	Low	4.4	free	Summer	V-neck	short	empire	cotton	Cordur

500 rows × 14 columns



```
In [31]: cursor.execute('SELECT * FROM ineuron.Dress_Sales_Data')
table_rows1=cursor.fetchall()
Dress_Sales_df = pd.DataFrame(table_rows1)
Dress_Sales_df
```

```
Out[31]:
```

	0	1	2	3	4	5	6	7	8	9	...	14	15	16	
0	1006032852	2114	2274	2491	2660	2727	2887	2930	3119	3204	...	3554	3624	3706	37
1	1212192089	151	275	570	750	813	1066	1164	1558	1756	...	2710	2942	3258	33
2	1190380701	6	7	7	7	8	8	9	10	10	...	11	11	11	
3	966005983	1005	1128	1326	1455	1507	1621	1637	1723	1746	...	1878	1892	1914	19
4	876339541	996	1175	1304	1396	1432	1559	1570	1638	1655	...	2032	2156	2252	23
...
495	713391965	0	0	0	560	554	544	537	525	519	...	400	388	360	3
496	722565148	0	0	0	875	866	861	854	850	844	...	859	866	882	8
497	532874347	0	0	0	734	728	726	715	694	690	...	616	597	586	5
498	655464934	0	0	0	254	259	261	263	268	270	...	257	256	255	2
499	919930954	0	0	0	538	545	558	563	578	585	...	628	632	639	6

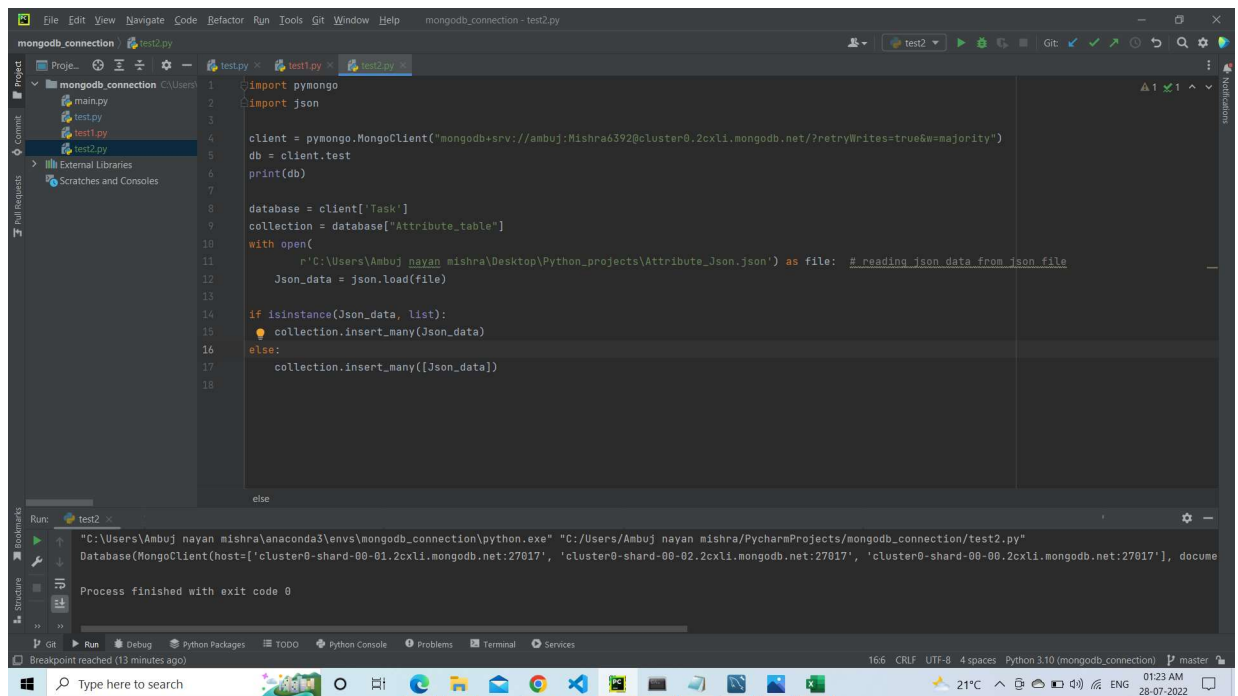
500 rows × 24 columns



4. Convert attribute dataset in json format

```
In [34]: Attribute_df.to_json('Attribute_Json.json') # data storing
```

5.Store this json dataset into mongodb

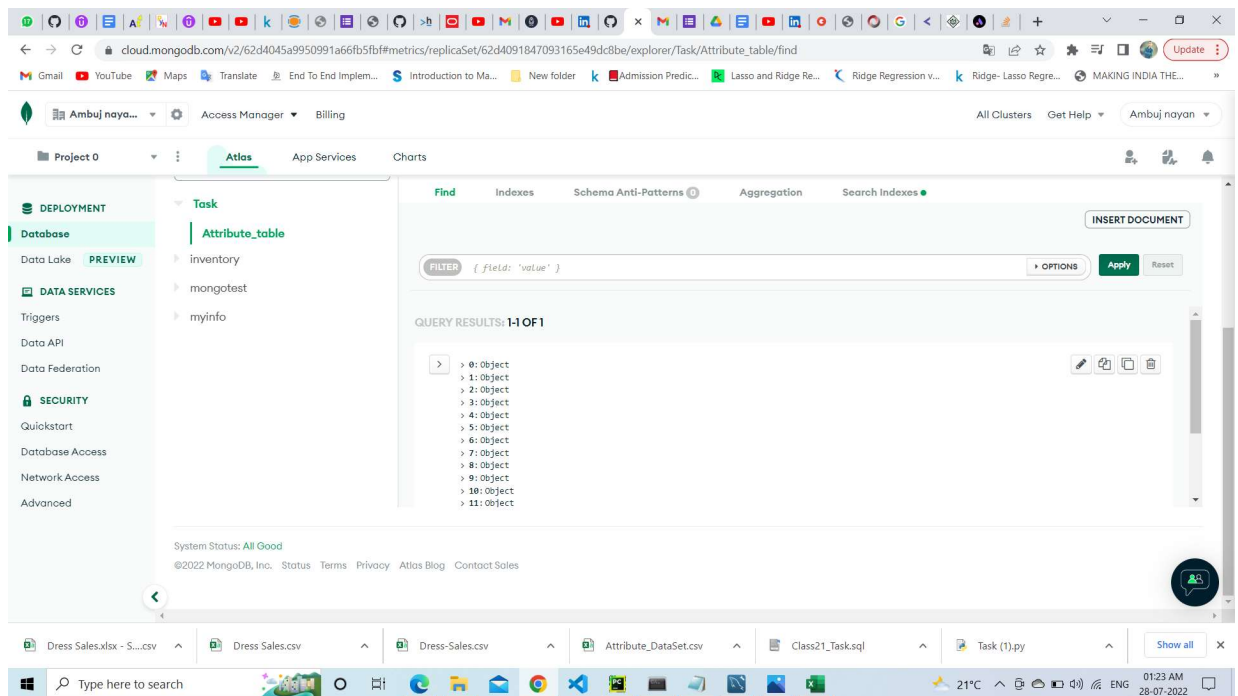


```
import pymongo
import json

client =
pymongo.MongoClient("mongodb+srv://ambuj:Mishra6392@cluster0.2cxli.mongodb.net/
?retryWrites=true&w=majority")
db = client.test
print(db)

database = client['Task']
collection = database["Attribute_table"]
with open(
    r'C:\Users\Ambuj nayan
mishra\Desktop\Python_projects\Attribute.Json.json') as file: # reading json
data from json file
    Json_data = json.load(file)

if isinstance(Json_data, list):
    collection.insert_many(Json_data)
else:
    collection.insert_many([Json_data])
```



6. in sql task try to perform left join operation with attrubute dataset and dress dataset on column Dress_ID

```
select Attribute_Data.Dress_ID,Style, Price, Rating, Size,
Season,NeckLine,SleeveLength
from Attribute_Data
left join Dress_Sales_Data ON Attribute_Data.Dress_ID =
Dress_Sales_Data.Dress_ID; # left join operation
```

7. Write a sql query to find out how many unique dress that we have based on dress id

```
SELECT count(DISTINCT(Dress_ID)) FROM Attribute_Data; # 475 unique dress
```

8. Try to find out how mnay dress is having recommendation 0

```
SELECT count(Recommendation) FROM Attribute_Data where Recommendation=0; # 290
recommendation 0
```

9.Try to find out total dress sell for individual dress id

```
select Dress_ID,(`29-08-2013` + `31-08-2013` + `02-09-2013` + `04-09-2013` +
`06-09-2013` + `08-09-2013` + `10-09-2013` +
```

```
`12-09-2013` + `14-09-2013` + `16-09-2013` + `18-09-2013` + `20-09-2013` +
`22-09-2013` + `24-09-2013` + `26-09-2013` +
`28-09-2013`+`30-09-2013`+`02-10-2013`+`04-10-2013`+`06-10-2013`+`08-10-
2010`+`10-10-2013`+`12-10-2013`) as total from Dress_Sales_Data
```

```
Dress_ID      total sell
'1006032852', '75979'
'1212192089', '52256'
'1190380701', '223'
'966005983', '39691'
'876339541', '44077'
'1068332458', '457'
'1220707172', '7328'
'1219677488', '3248'
'1113094204', '494'
'985292672', '300'
'1117293701', '1911'
'898481530', '2842'
'957723897', '49983'
'749031896', '75346'
'1055411544', '1095'
'1162628131', '2319'
'624314841', '47998'
'830467746', '401'
'840857118', '299'
'1113221101', '9998'
'861754372', '6175'
'856178100', '31467'
'1122989777', '5216'
'840516484', '42347'
'768517084', '107'
-----
-----
-----
```

10. Try to find out a third highest most selling dress id

```
select Dress_ID,(`29-08-2013` + `31-08-2013` + `02-09-2013` + `04-09-2013` +
`06-09-2013` + `08-09-2013` + `10-09-2013` +
`12-09-2013` + `14-09-2013` + `16-09-2013` + `18-09-2013` + `20-09-2013` +
`22-09-2013` + `24-09-2013` + `26-09-2013` +
`28-09-2013`+`30-09-2013`+`02-10-2013`+`04-10-2013`+`06-10-2013`+`08-10-
2010`+`10-10-2013`+`12-10-2013`) as total from Dress_Sales_Data ORDER BY total
DESC limit 1 offset 2

#Dress_ID,      total
# '1006032852', '75979'
```

In []:

