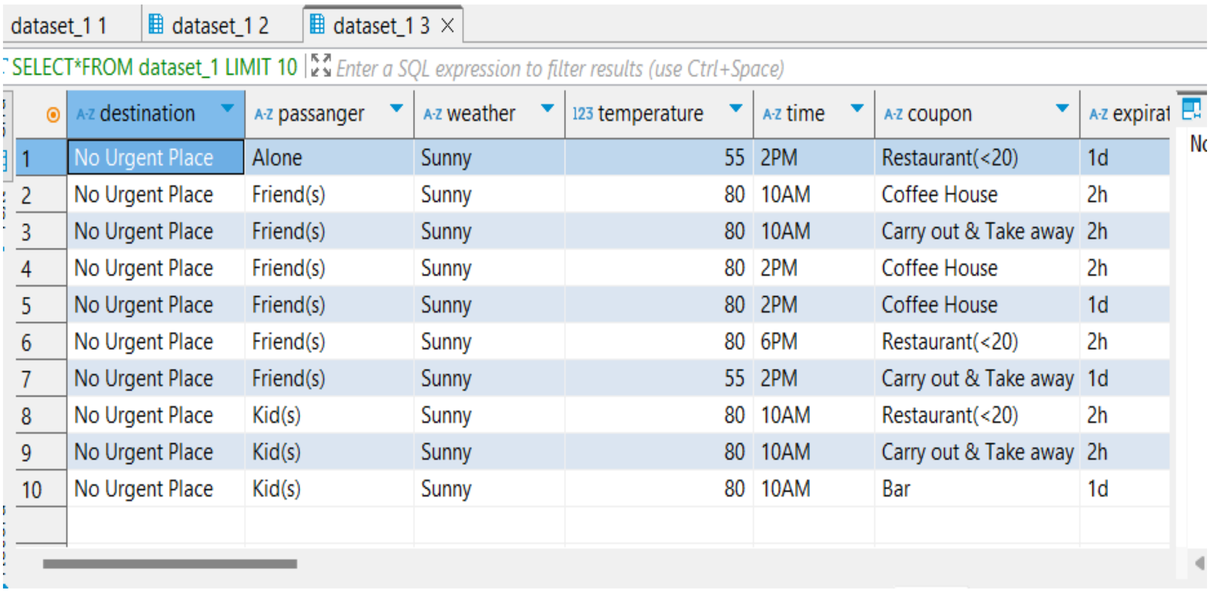
SQL vs PYTHON DATA ANALYSIS

SQL Query

Select * from dataset_1;

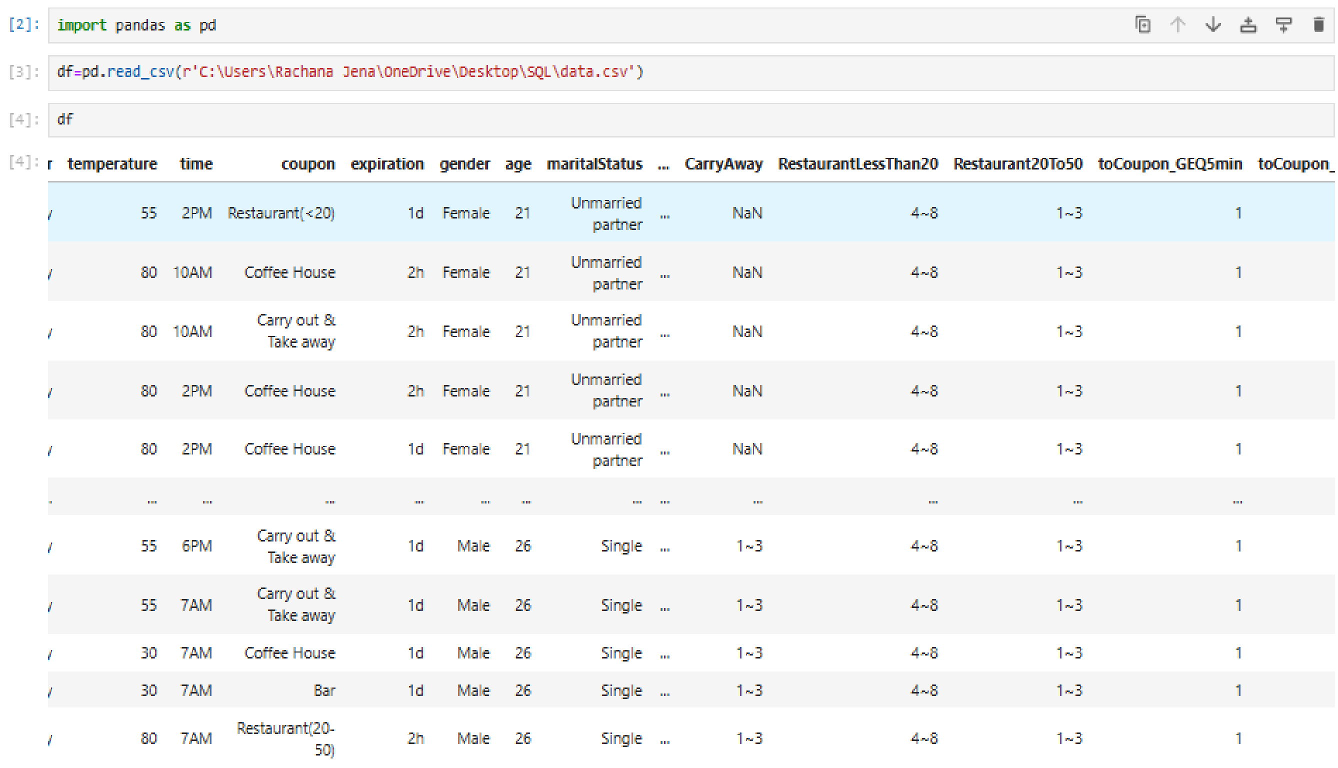


```
SELECT*FROM dataset_1 LIMIT 10
```

	destination	passanger	weather	temperature	time	coupon	expirat
1	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1d
2	No Urgent Place	Friend(s)	Sunny	80	10AM	Coffee House	2h
3	No Urgent Place	Friend(s)	Sunny	80	10AM	Carry out & Take away	2h
4	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	2h
5	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	1d
6	No Urgent Place	Friend(s)	Sunny	80	6PM	Restaurant(<20)	2h
7	No Urgent Place	Friend(s)	Sunny	55	2PM	Carry out & Take away	1d
8	No Urgent Place	Kid(s)	Sunny	80	10AM	Restaurant(<20)	2h
9	No Urgent Place	Kid(s)	Sunny	80	10AM	Carry out & Take away	2h
10	No Urgent Place	Kid(s)	Sunny	80	10AM	Bar	1d

Python Code

```
df
```



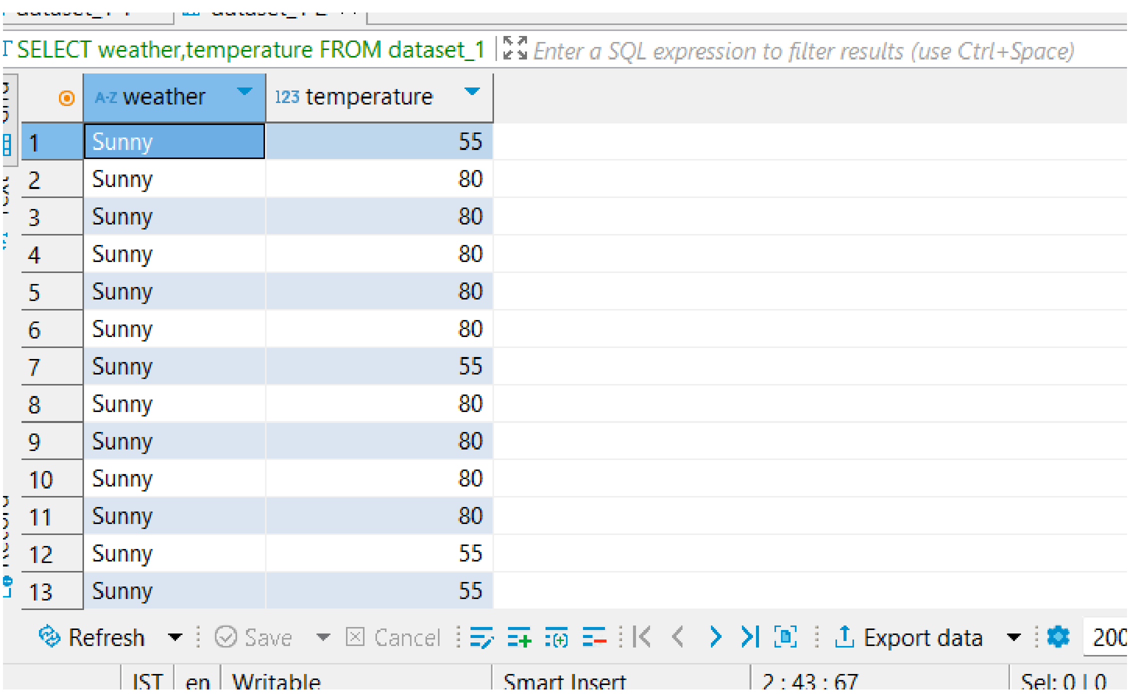
```
[2]: import pandas as pd

[3]: df=pd.read_csv(r'C:\Users\Rachana Jena\OneDrive\Desktop\SQL\data.csv')

[4]: df
```

[4]:

r	temperature	time	coupon	expiration	gender	age	maritalStatus	...	CarryAway	RestaurantLessThan20	Restaurant20To50	toCoupon_GEQ5min	toCoupon_
/	55	2PM	Restaurant(<20)	1d	Female	21	Unmarried partner	...	NaN	4~8	1~3	1	
/	80	10AM	Coffee House	2h	Female	21	Unmarried partner	...	NaN	4~8	1~3	1	
/	80	10AM	Carry out & Take away	2h	Female	21	Unmarried partner	...	NaN	4~8	1~3	1	
/	80	2PM	Coffee House	2h	Female	21	Unmarried partner	...	NaN	4~8	1~3	1	
/	80	2PM	Coffee House	1d	Female	21	Unmarried partner	...	NaN	4~8	1~3	1	
.	...	...	...	...	...	...	...	...	...	...	...	...	
/	55	6PM	Carry out & Take away	1d	Male	26	Single	...	1~3	4~8	1~3	1	
/	55	7AM	Carry out & Take away	1d	Male	26	Single	...	1~3	4~8	1~3	1	
/	30	7AM	Coffee House	1d	Male	26	Single	...	1~3	4~8	1~3	1	
/	30	7AM	Bar	1d	Male	26	Single	...	1~3	4~8	1~3	1	
/	80	7AM	Restaurant(20-50)	2h	Male	26	Single	...	1~3	4~8	1~3	1	

SQL Query

SELECT weather, temperature FROM dataset_1

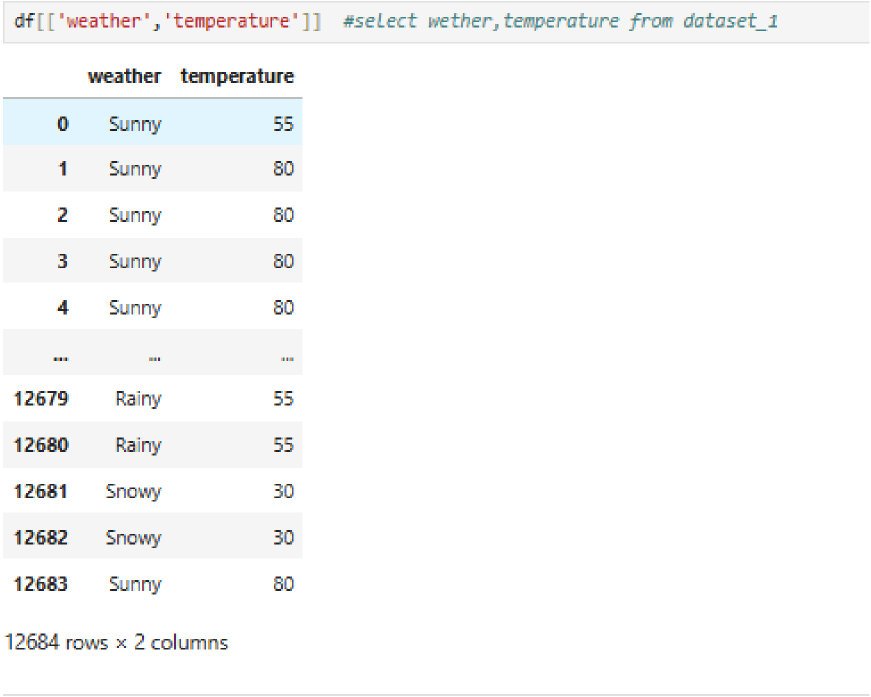


```
SELECT weather,temperature FROM dataset_1
```

	weather	temperature
1	Sunny	55
2	Sunny	80
3	Sunny	80
4	Sunny	80
5	Sunny	80
6	Sunny	80
7	Sunny	55
8	Sunny	80
9	Sunny	80
10	Sunny	80
11	Sunny	80
12	Sunny	55
13	Sunny	55

Python Code

df[[‘weather’, ‘temperature’]]



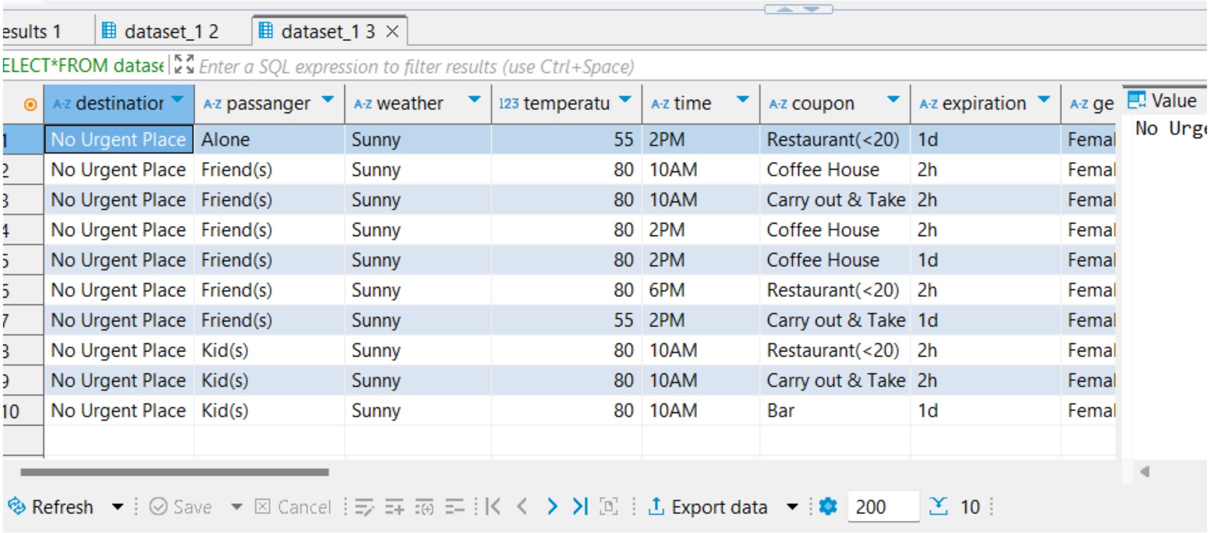
```
df[['weather','temperature']]   #select wether,temperature from dataset_1
```

	weather	temperature
0	Sunny	55
1	Sunny	80
2	Sunny	80
3	Sunny	80
4	Sunny	80
...	...	...
12679	Rainy	55
12680	Rainy	55
12681	Snowy	30
12682	Snowy	30
12683	Sunny	80

12684 rows × 2 columns

SQL Query

SELECT*FROM dataset_1 LIMIT 10



ELECT*FROM datase Enter a SQL expression to filter results (use Ctrl+Space)

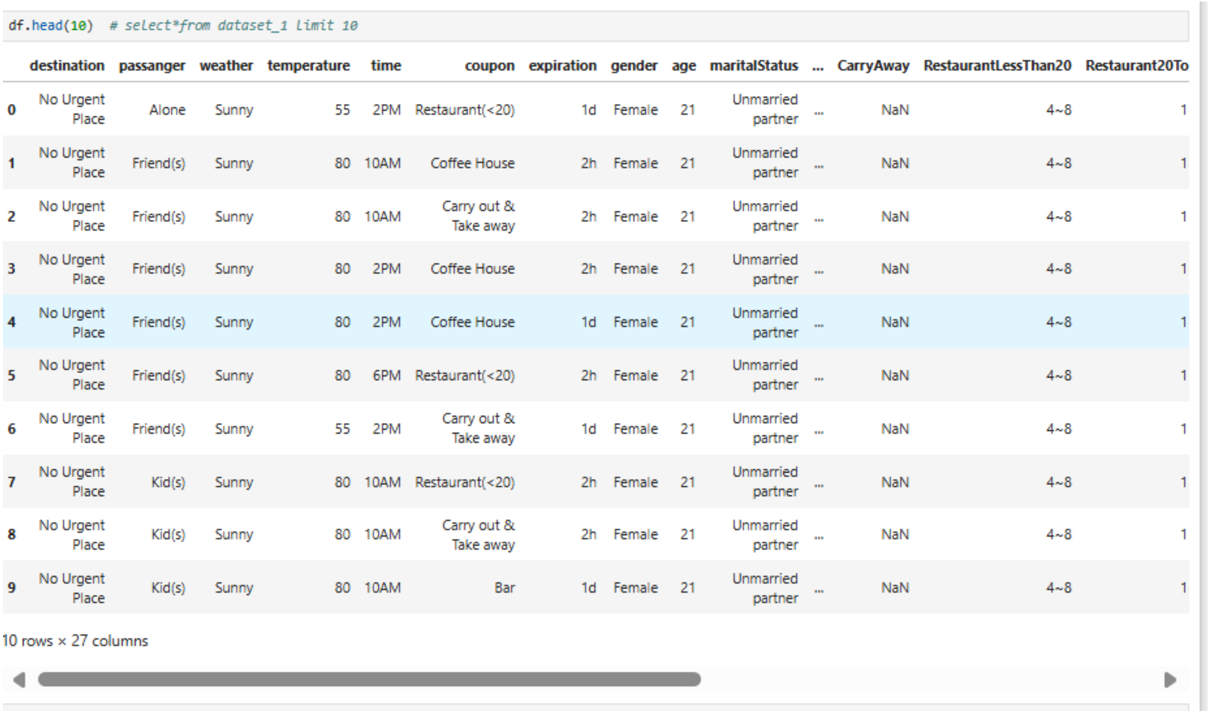
	destinatior	passanger	weather	temperatu	time	coupon	expiration	ge
1	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1d	Femal
2	No Urgent Place	Friend(s)	Sunny	80	10AM	Coffee House	2h	Femal
3	No Urgent Place	Friend(s)	Sunny	80	10AM	Carry out & Take	2h	Femal
4	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	2h	Femal
5	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	1d	Femal
5	No Urgent Place	Friend(s)	Sunny	80	6PM	Restaurant(<20)	2h	Femal
7	No Urgent Place	Friend(s)	Sunny	55	2PM	Carry out & Take	1d	Femal
3	No Urgent Place	Kid(s)	Sunny	80	10AM	Restaurant(<20)	2h	Femal
9	No Urgent Place	Kid(s)	Sunny	80	10AM	Carry out & Take	2h	Femal
10	No Urgent Place	Kid(s)	Sunny	80	10AM	Bar	1d	Femal

No Urge

200 10

Python Code

Df[[‘weather’, ‘temperature’]]



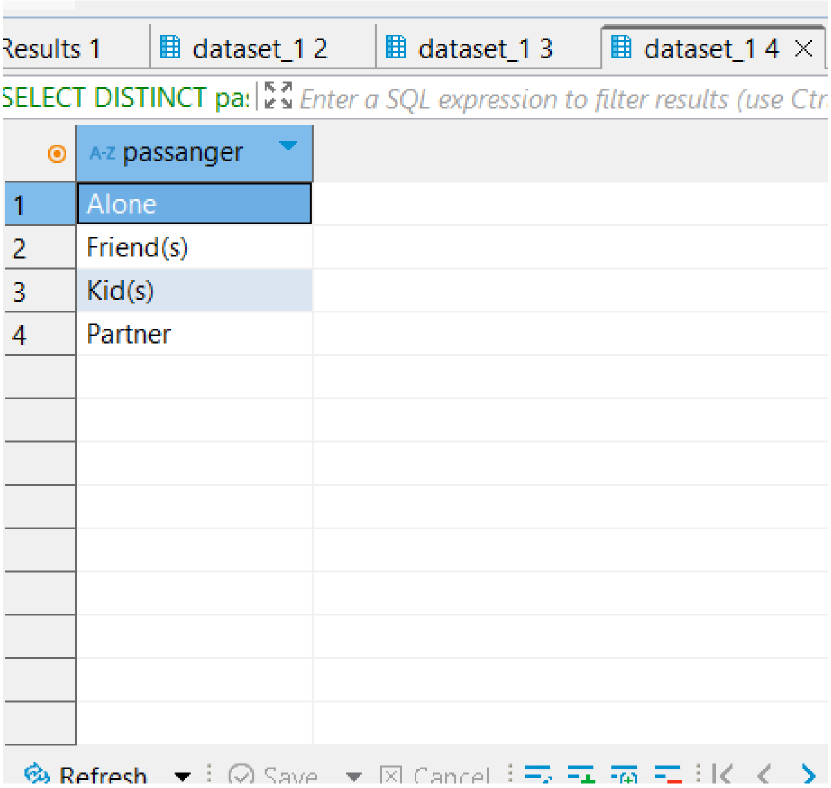
```
df.head(10)  # select*from dataset_1 limit 10
```

	destination	passanger	weather	temperature	time	coupon	expiration	gender	age	maritalStatus	...	CarryAway	RestaurantLessThan20	Restaurant20To
0	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1d	Female	21	Unmarried partner	...	NaN	4~8	1
1	No Urgent Place	Friend(s)	Sunny	80	10AM	Coffee House	2h	Female	21	Unmarried partner	...	NaN	4~8	1
2	No Urgent Place	Friend(s)	Sunny	80	10AM	Carry out & Take away	2h	Female	21	Unmarried partner	...	NaN	4~8	1
3	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	2h	Female	21	Unmarried partner	...	NaN	4~8	1
4	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	1d	Female	21	Unmarried partner	...	NaN	4~8	1
5	No Urgent Place	Friend(s)	Sunny	80	6PM	Restaurant(<20)	2h	Female	21	Unmarried partner	...	NaN	4~8	1
6	No Urgent Place	Friend(s)	Sunny	55	2PM	Carry out & Take away	1d	Female	21	Unmarried partner	...	NaN	4~8	1
7	No Urgent Place	Kid(s)	Sunny	80	10AM	Restaurant(<20)	2h	Female	21	Unmarried partner	...	NaN	4~8	1
8	No Urgent Place	Kid(s)	Sunny	80	10AM	Carry out & Take away	2h	Female	21	Unmarried partner	...	NaN	4~8	1
9	No Urgent Place	Kid(s)	Sunny	80	10AM	Bar	1d	Female	21	Unmarried partner	...	NaN	4~8	1

10 rows × 27 columns

SQL Query

SELECT DISTINCT passenger FROM dataset_1

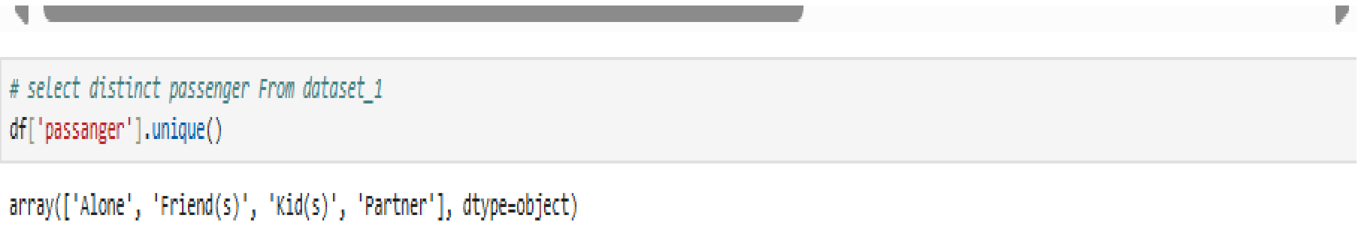


SELECT DISTINCT pa: Enter a SQL expression to filter results (use Ctr

	passanger
1	Alone
2	Friend(s)
3	Kid(s)
4	Partner

Python Code

df[‘passenger’].unique()

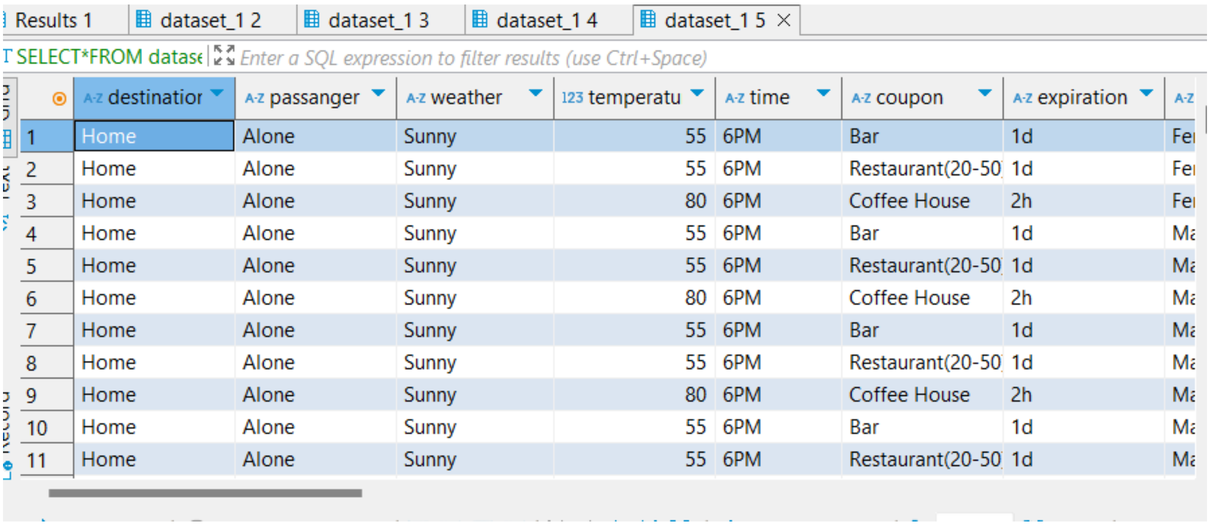


```
# select distinct passenger From dataset_1
df['passanger'].unique()
```

```
array(['Alone', 'Friend(s)', 'Kid(s)', 'Partner'], dtype=object)
```

SQL Query

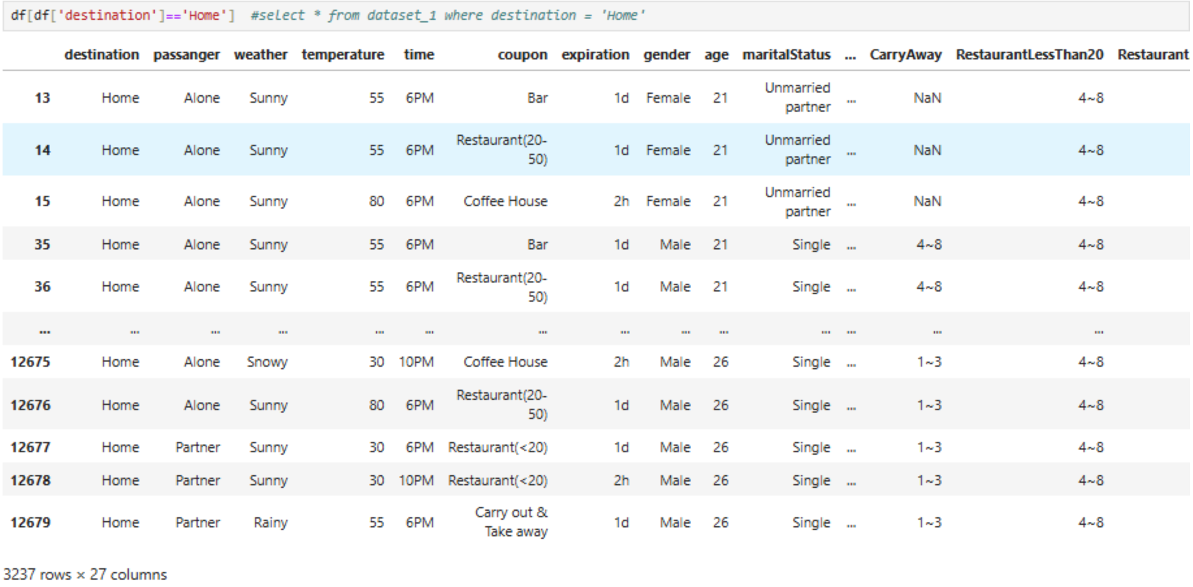
SELECT*FROM dataset_1 WHERE destination = ‘Home’



	destinatior	passanger	weather	temperatu	time	coupon	expiration	
1	Home	Alone	Sunny	55	6PM	Bar	1d	Fe
2	Home	Alone	Sunny	55	6PM	Restaurant(20-50	1d	Fe
3	Home	Alone	Sunny	80	6PM	Coffee House	2h	Fe
4	Home	Alone	Sunny	55	6PM	Bar	1d	Ma
5	Home	Alone	Sunny	55	6PM	Restaurant(20-50	1d	Ma
6	Home	Alone	Sunny	80	6PM	Coffee House	2h	Ma
7	Home	Alone	Sunny	55	6PM	Bar	1d	Ma
8	Home	Alone	Sunny	55	6PM	Restaurant(20-50	1d	Ma
9	Home	Alone	Sunny	80	6PM	Coffee House	2h	Ma
10	Home	Alone	Sunny	55	6PM	Bar	1d	Ma
11	Home	Alone	Sunny	55	6PM	Restaurant(20-50	1d	Ma

Python Code

Df[df[‘destination’]==’Home’]



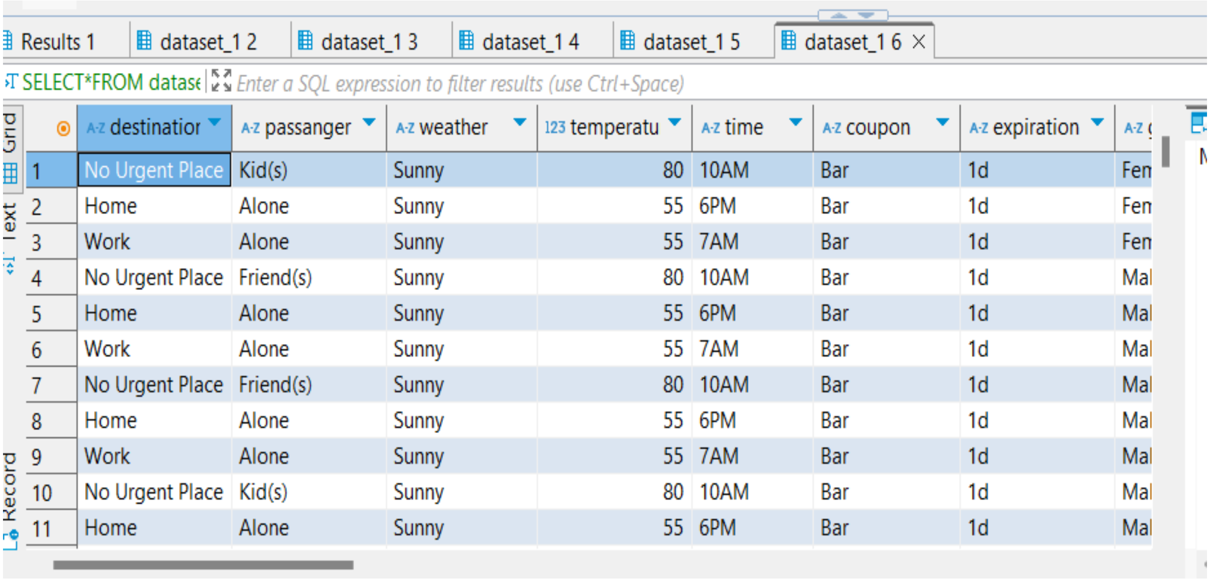
```
df[df['destination']=='Home']  #select * from dataset_1 where destination = 'Home'
```

	destination	passanger	weather	temperature	time	coupon	expiration	gender	age	maritalStatus	...	CarryAway	RestaurantLessThan20	Restaurant
13	Home	Alone	Sunny	55	6PM	Bar	1d	Female	21	Unmarried partner	...	NaN	4~8	
14	Home	Alone	Sunny	55	6PM	Restaurant(20-50)	1d	Female	21	Unmarried partner	...	NaN	4~8	
15	Home	Alone	Sunny	80	6PM	Coffee House	2h	Female	21	Unmarried partner	...	NaN	4~8	
35	Home	Alone	Sunny	55	6PM	Bar	1d	Male	21	Single	...	4~8	4~8	
36	Home	Alone	Sunny	55	6PM	Restaurant(20-50)	1d	Male	21	Single	...	4~8	4~8	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
12675	Home	Alone	Snowy	30	10PM	Coffee House	2h	Male	26	Single	...	1~3	4~8	
12676	Home	Alone	Sunny	80	6PM	Restaurant(20-50)	1d	Male	26	Single	...	1~3	4~8	
12677	Home	Partner	Sunny	30	6PM	Restaurant(<20)	1d	Male	26	Single	...	1~3	4~8	
12678	Home	Partner	Sunny	30	10PM	Restaurant(<20)	2h	Male	26	Single	...	1~3	4~8	
12679	Home	Partner	Rainy	55	6PM	Carry out & Take away	1d	Male	26	Single	...	1~3	4~8	

3237 rows × 27 columns

SQL Query

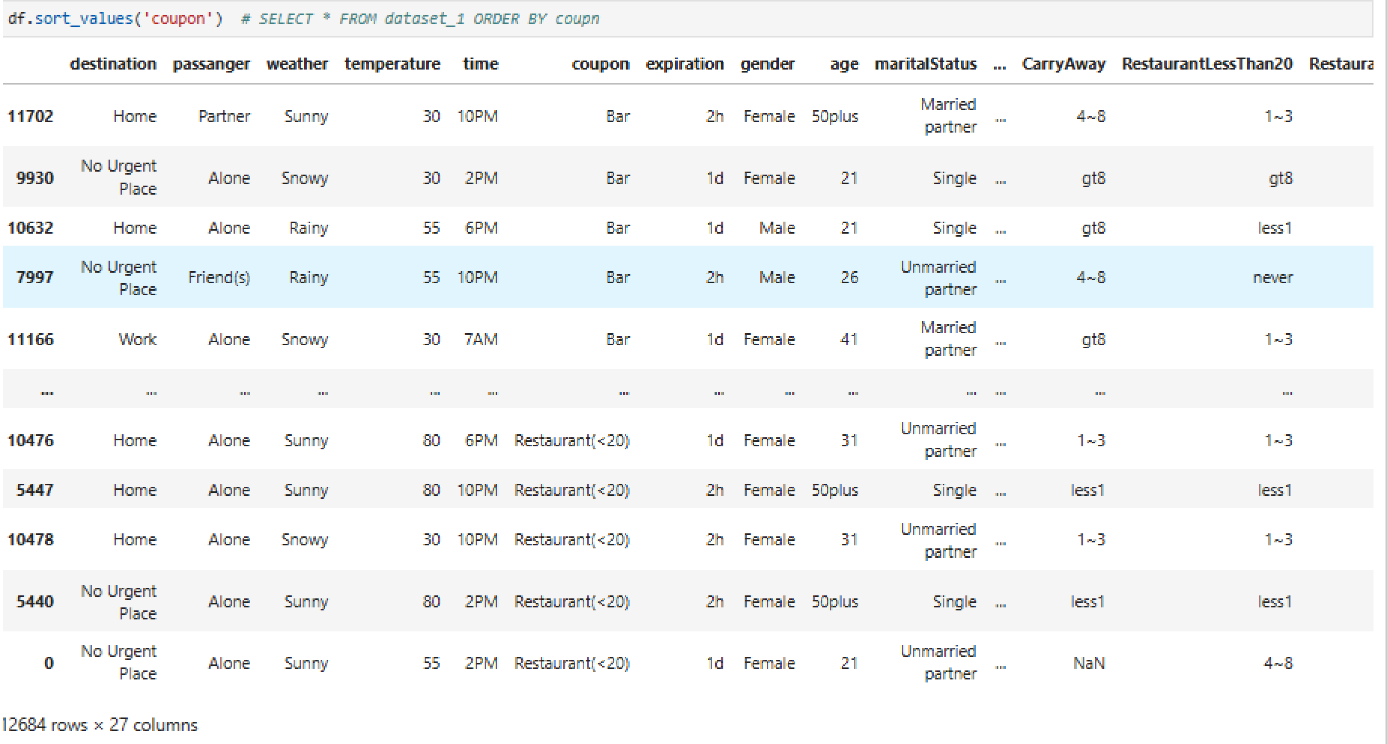
SELECT*FROM dataset_1 ORDER BY coupon



	destinatior	passanger	weather	temperatu	time	coupon	expiration	A-Z g
1	No Urgent Place	Kid(s)	Sunny	80	10AM	Bar	1d	Fen
2	Home	Alone	Sunny	55	6PM	Bar	1d	Fen
3	Work	Alone	Sunny	55	7AM	Bar	1d	Fen
4	No Urgent Place	Friend(s)	Sunny	80	10AM	Bar	1d	Mal
5	Home	Alone	Sunny	55	6PM	Bar	1d	Mal
6	Work	Alone	Sunny	55	7AM	Bar	1d	Mal
7	No Urgent Place	Friend(s)	Sunny	80	10AM	Bar	1d	Mal
8	Home	Alone	Sunny	55	6PM	Bar	1d	Mal
9	Work	Alone	Sunny	55	7AM	Bar	1d	Mal
10	No Urgent Place	Kid(s)	Sunny	80	10AM	Bar	1d	Mal
11	Home	Alone	Sunny	55	6PM	Bar	1d	Mal

Python Code

df. sort_values('coupon')



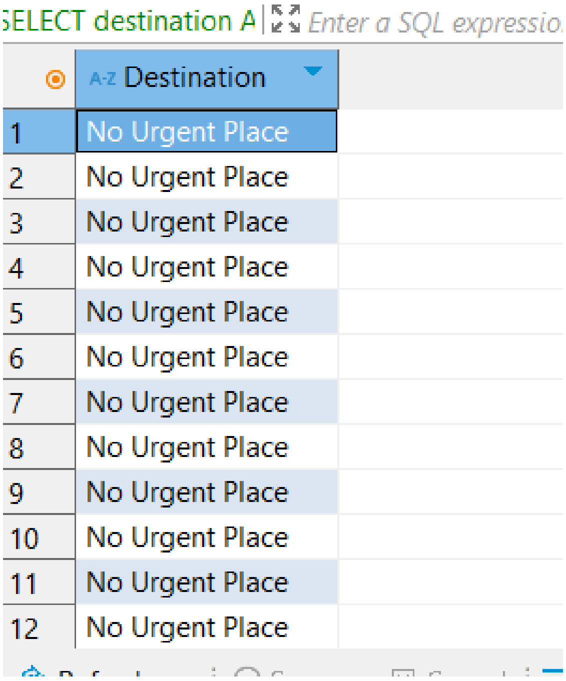
```
df.sort_values('coupon')  # SELECT * FROM dataset_1 ORDER BY coupn
```

	destination	passanger	weather	temperature	time	coupon	expiration	gender	age	maritalStatus	...	CarryAway	RestaurantLessThan20	Restaura
11702	Home	Partner	Sunny	30	10PM	Bar	2h	Female	50plus	Married partner	...	4~8	1~3	
9930	No Urgent Place	Alone	Snowy	30	2PM	Bar	1d	Female	21	Single	...	gt8	gt8	
10632	Home	Alone	Rainy	55	6PM	Bar	1d	Male	21	Single	...	gt8	less1	
7997	No Urgent Place	Friend(s)	Rainy	55	10PM	Bar	2h	Male	26	Unmarried partner	...	4~8	never	
11166	Work	Alone	Snowy	30	7AM	Bar	1d	Female	41	Married partner	...	gt8	1~3	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
10476	Home	Alone	Sunny	80	6PM	Restaurant(<20)	1d	Female	31	Unmarried partner	...	1~3	1~3	
5447	Home	Alone	Sunny	80	10PM	Restaurant(<20)	2h	Female	50plus	Single	...	less1	less1	
10478	Home	Alone	Snowy	30	10PM	Restaurant(<20)	2h	Female	31	Unmarried partner	...	1~3	1~3	
5440	No Urgent Place	Alone	Sunny	80	2PM	Restaurant(<20)	2h	Female	50plus	Single	...	less1	less1	
0	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1d	Female	21	Unmarried partner	...	NaN	4~8	

12684 rows × 27 columns

SQL Query

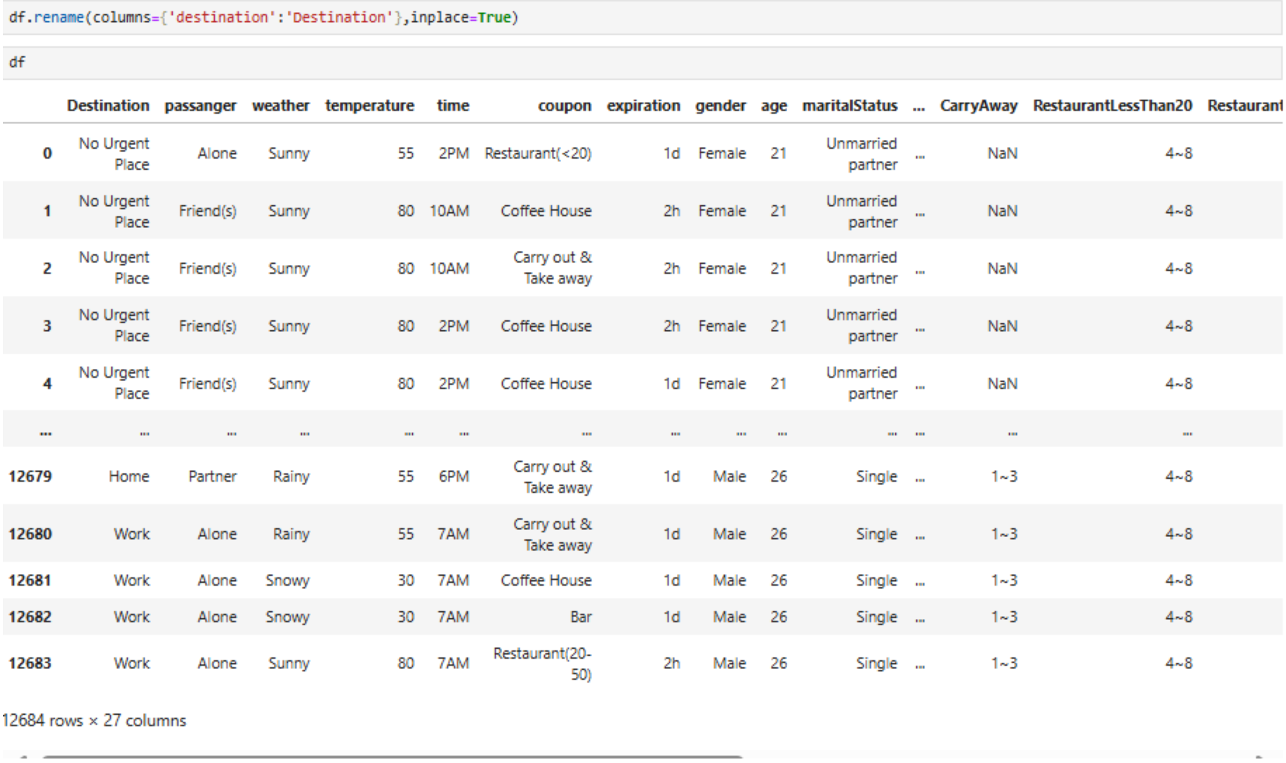
SELECT destination as Destination FROM dataset_1



	Destination
1	No Urgent Place
2	No Urgent Place
3	No Urgent Place
4	No Urgent Place
5	No Urgent Place
6	No Urgent Place
7	No Urgent Place
8	No Urgent Place
9	No Urgent Place
10	No Urgent Place
11	No Urgent Place
12	No Urgent Place

Python Code

df. rename(columns={‘destination’: ‘Destination’},inplace=True)



```
df.rename(columns={'destination':'Destination'},inplace=True)
```

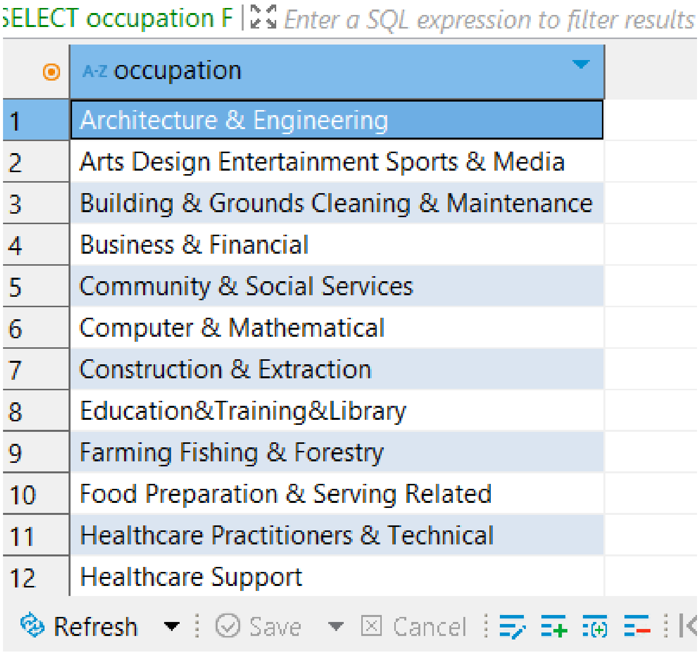
```
df
```

	Destination	passanger	weather	temperature	time	coupon	expiration	gender	age	maritalStatus	...	CarryAway	RestaurantLessThan20	Restaurant
0	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1d	Female	21	Unmarried partner	...	NaN	4~8	
1	No Urgent Place	Friend(s)	Sunny	80	10AM	Coffee House	2h	Female	21	Unmarried partner	...	NaN	4~8	
2	No Urgent Place	Friend(s)	Sunny	80	10AM	Carry out & Take away	2h	Female	21	Unmarried partner	...	NaN	4~8	
3	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	2h	Female	21	Unmarried partner	...	NaN	4~8	
4	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	1d	Female	21	Unmarried partner	...	NaN	4~8	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
12679	Home	Partner	Rainy	55	6PM	Carry out & Take away	1d	Male	26	Single	...	1~3	4~8	
12680	Work	Alone	Rainy	55	7AM	Carry out & Take away	1d	Male	26	Single	...	1~3	4~8	
12681	Work	Alone	Snowy	30	7AM	Coffee House	1d	Male	26	Single	...	1~3	4~8	
12682	Work	Alone	Snowy	30	7AM	Bar	1d	Male	26	Single	...	1~3	4~8	
12683	Work	Alone	Sunny	80	7AM	Restaurant(20-50)	2h	Male	26	Single	...	1~3	4~8	

12684 rows × 27 columns

SQL Query

SELECT occupation FROM dataset_1 GROUP BY occupation

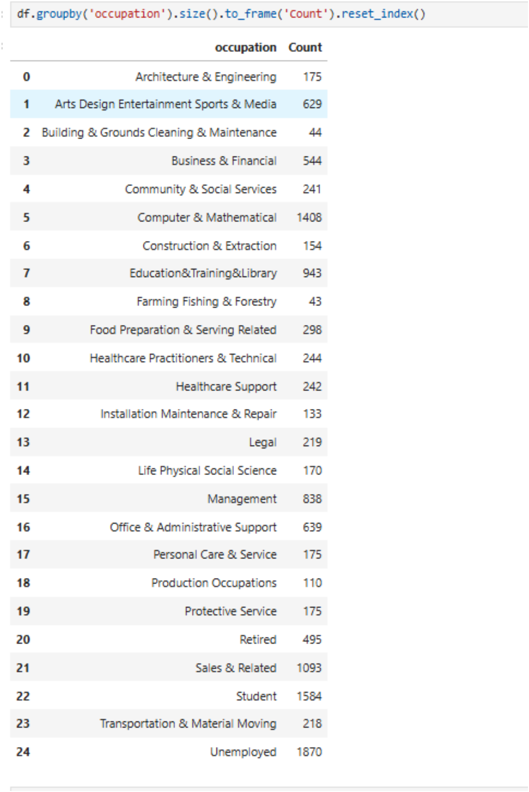


```
;ELECT occupation F
```

	occupation
1	Architecture & Engineering
2	Arts Design Entertainment Sports & Media
3	Building & Grounds Cleaning & Maintenance
4	Business & Financial
5	Community & Social Services
6	Computer & Mathematical
7	Construction & Extraction
8	Education&Training&Library
9	Farming Fishing & Forestry
10	Food Preparation & Serving Related
11	Healthcare Practitioners & Technical
12	Healthcare Support

Python Code

df. groupby(‘occupation’).size().to_frame(‘Count’).reset_index()

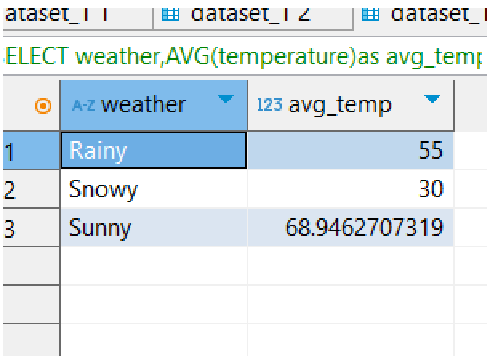


```
df.groupby('occupation').size().to_frame('Count').reset_index()
```

	occupation	Count
0	Architecture & Engineering	175
1	Arts Design Entertainment Sports & Media	629
2	Building & Grounds Cleaning & Maintenance	44
3	Business & Financial	544
4	Community & Social Services	241
5	Computer & Mathematical	1408
6	Construction & Extraction	154
7	Education&Training&Library	943
8	Farming Fishing & Forestry	43
9	Food Preparation & Serving Related	298
10	Healthcare Practitioners & Technical	244
11	Healthcare Support	242
12	Installation Maintenance & Repair	133
13	Legal	219
14	Life Physical Social Science	170
15	Management	838
16	Office & Administrative Support	639
17	Personal Care & Service	175
18	Production Occupations	110
19	Protective Service	175
20	Retired	495
21	Sales & Related	1093
22	Student	1584
23	Transportation & Material Moving	218
24	Unemployed	1870

SQL Query

SELECT weather, AVG (temperature) as avg_temp FROM dataset_1 GROUP BY weather



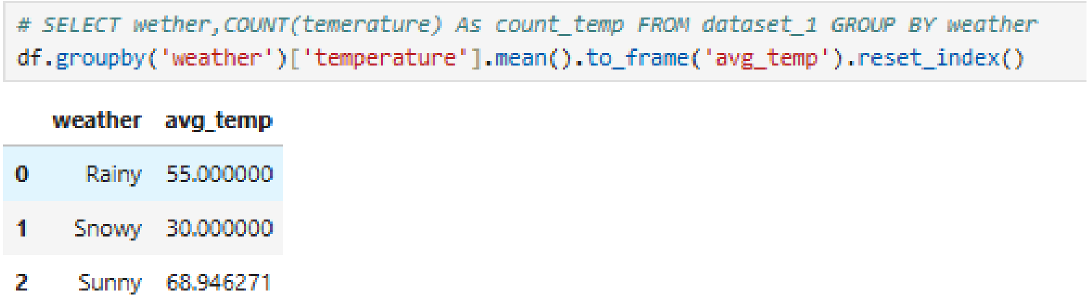
ataset_1 1 | dataset_1 2 | dataset_

```
ELECT weather,AVG(temperature)as avg_temp
```

	weather	avg_temp
1	Rainy	55
2	Snowy	30
3	Sunny	68.9462707319

Python Code

df.groupby('weather')['temperature'].mean().to_frame('avg_temp') .reset_index()

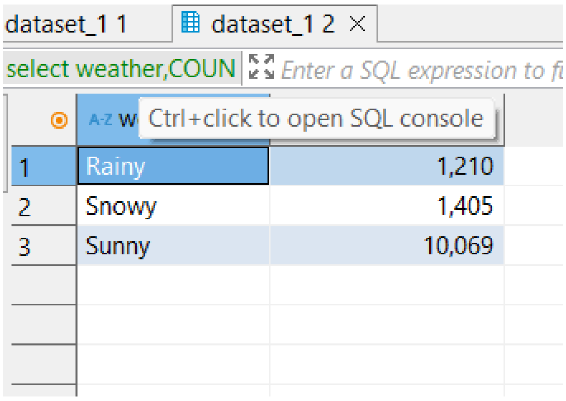


```
# SELECT wether,COUNT(temerature) As count_temp FROM dataset_1 GROUP BY weather
df.groupby('weather')['temperature'].mean().to_frame('avg_temp').reset_index()
```

	weather	avg_temp
0	Rainy	55.000000
1	Snowy	30.000000
2	Sunny	68.946271

SQL Query

SELECT weather, COUNT (temperature)AS count_temp FROM dataset_1 GROUP BY weather

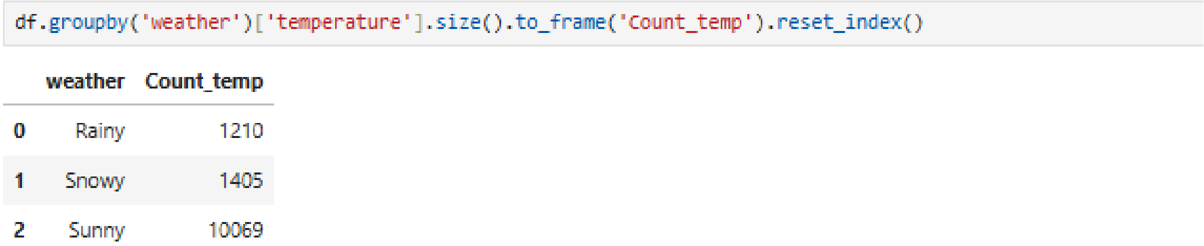


select weather,COUN

	w	
1	Rainy	1,210
2	Snowy	1,405
3	Sunny	10,069

Python Code

df.groupby('weather')['temperature'].size().to_frame('Count_temp').reset_index()

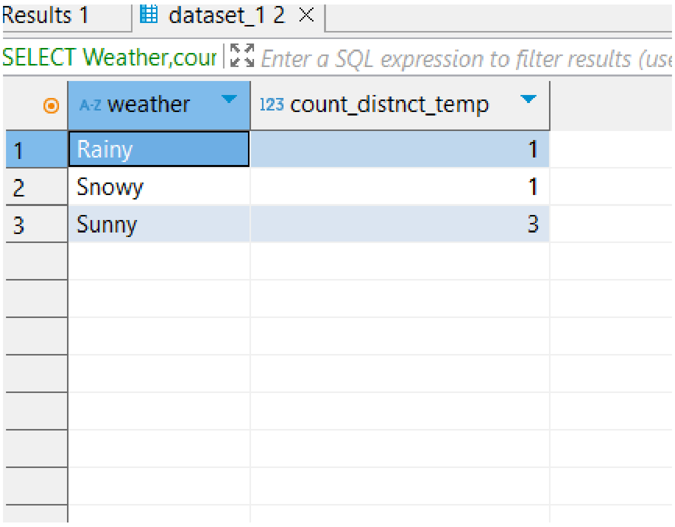


```
df.groupby('weather')['temperature'].size().to_frame('Count_temp').reset_index()
```

	weather	Count_temp
0	Rainy	1210
1	Snowy	1405
2	Sunny	10069

SQL Query

SELECT weather, COUNT (DISTINCT temperature) AS count_distinct_temp FROM dataset_1 GROUP BY weather

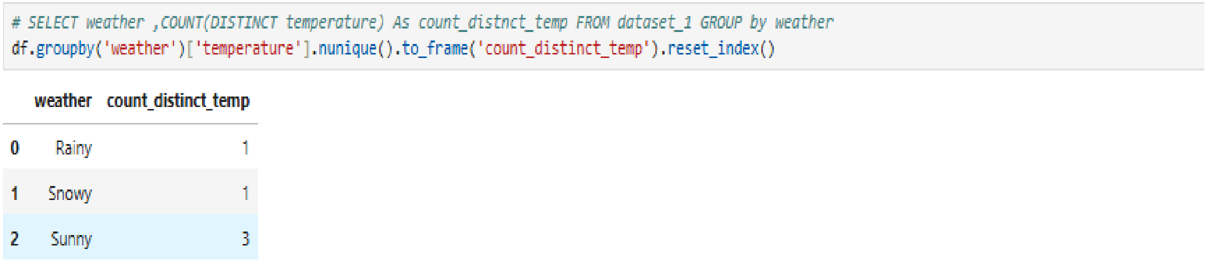


SELECT Weather,cour

	weather	count_distnct_temp
1	Rainy	1
2	Snowy	1
3	Sunny	3

Python Code

df.groupby('weather')['temperature'].nunique().to_frame('count_distinct_temp').reset_index()

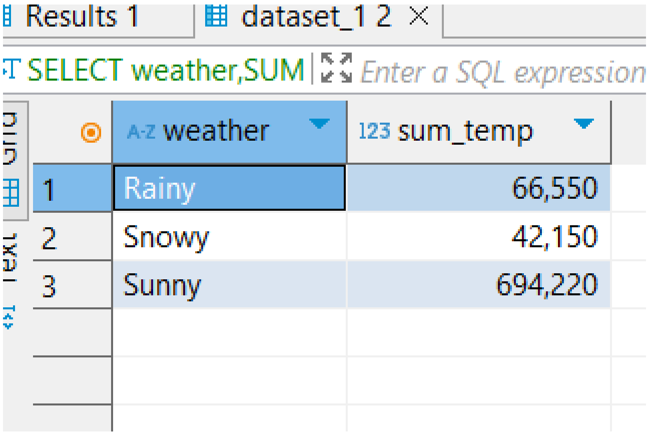


```
# SELECT weather ,COUNT(DISTINCT temperature) As count_distnct_temp FROM dataset_1 GROUP by weather
df.groupby('weather')['temperature'].nunique().to_frame('count_distinct_temp').reset_index()
```

	weather	count_distinct_temp
0	Rainy	1
1	Snowy	1
2	Sunny	3

SQL Query

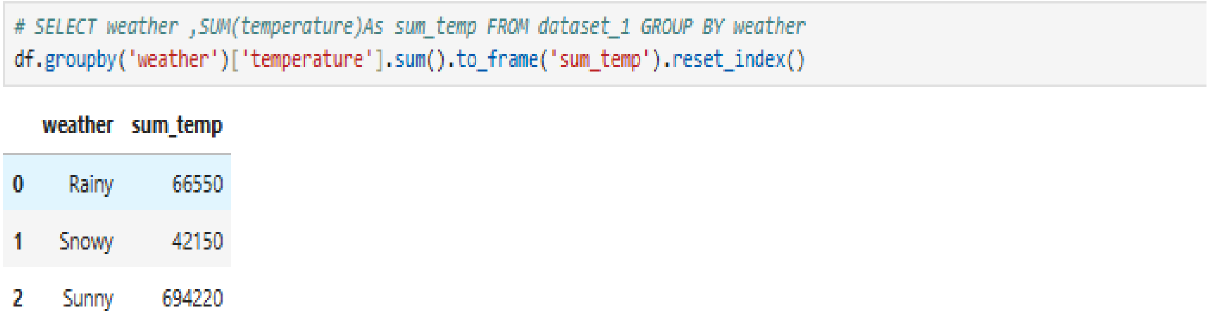
SELECT weather, SUM (temperature)As sum_temp FROM dataset_1 GROUP BY weather



	weather	sum_temp
1	Rainy	66,550
2	Snowy	42,150
3	Sunny	694,220

Python Code

df.groupby('weather')['temperature'].sum().to_frame('sum_temp').reset_index()

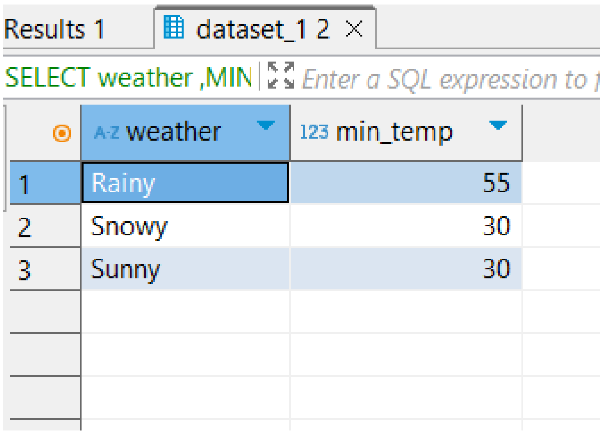


```
# SELECT weather ,SUM(temperature)As sum_temp FROM dataset_1 GROUP BY weather
df.groupby('weather')['temperature'].sum().to_frame('sum_temp').reset_index()
```

	weather	sum_temp
0	Rainy	66550
1	Snowy	42150
2	Sunny	694220

SQL Query

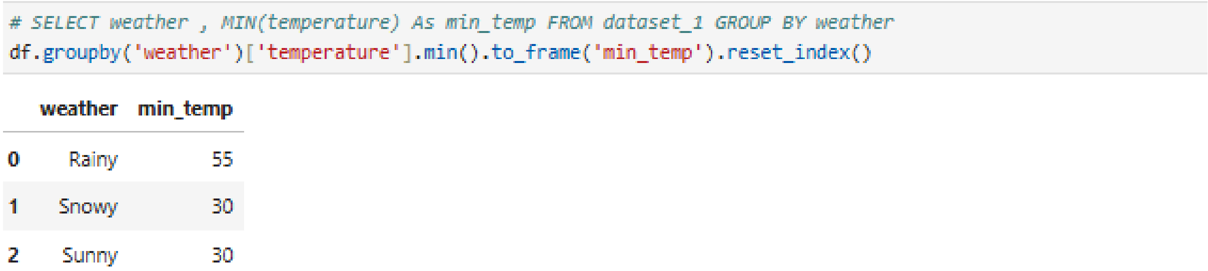
SELECT weather, MIN (temperature) AS min_temp FROM dataset_1 GROUP BY weather



	weather	min_temp
1	Rainy	55
2	Snowy	30
3	Sunny	30

Python Code

df.groupby(‘weather’)[‘temperature’].min().to_frame(‘min_temp’).reset_index()

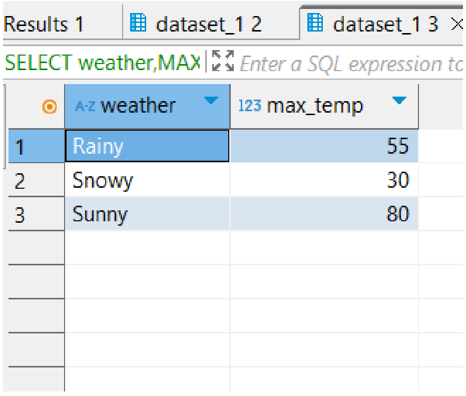


```
# SELECT weather , MIN(temperature) As min_temp FROM dataset_1 GROUP BY weather
df.groupby('weather')['temperature'].min().to_frame('min_temp').reset_index()
```

	weather	min_temp
0	Rainy	55
1	Snowy	30
2	Sunny	30

SQL Query

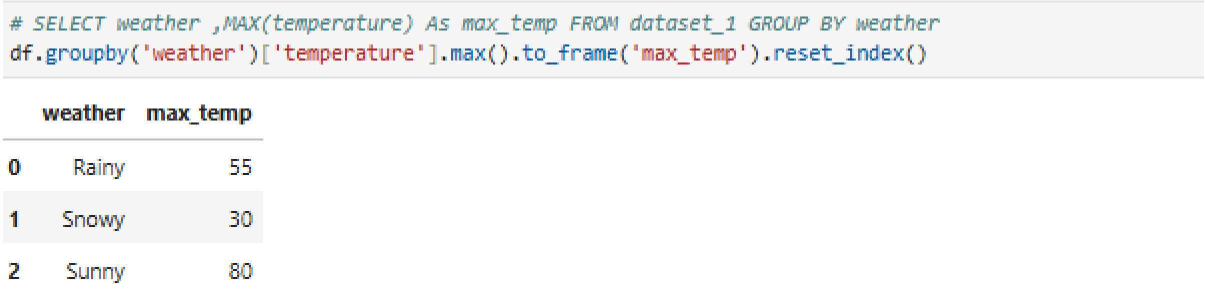
SELECT weather, MAX (temperature) AS max_temp FROM dataset_1 GROUP BY weather



	weather	max_temp
1	Rainy	55
2	Snowy	30
3	Sunny	80

Python Code

df.groupby(‘weather’)[‘temperature’].max().to_frame(‘max_temp’).reset_index()

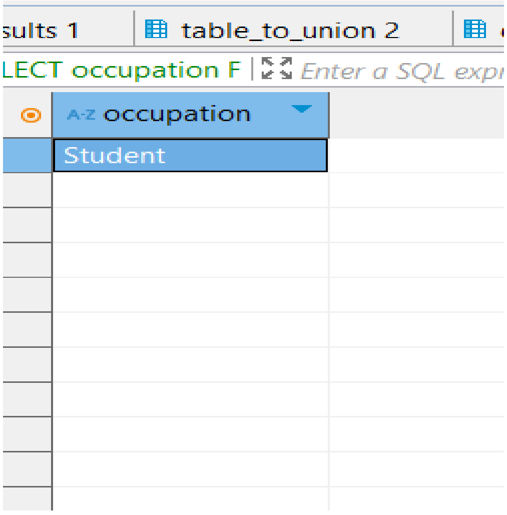


```
# SELECT weather ,MAX(temperature) As max_temp FROM dataset_1 GROUP BY weather
df.groupby('weather')['temperature'].max().to_frame('max_temp').reset_index()
```

	weather	max_temp
0	Rainy	55
1	Snowy	30
2	Sunny	80

SQL Query

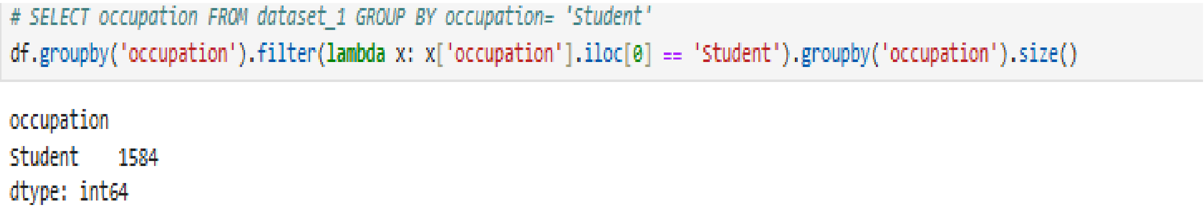
SELECT occupation FROM dataset_1 GROUP BY occupation HAVING occupation=’Student’



occupation
Student

Python Code

df. groupby(‘occupation’). filter (lambda x:x[‘occupation’]. iloc [0] ==’Student’). groupby(‘occupation’). size ()

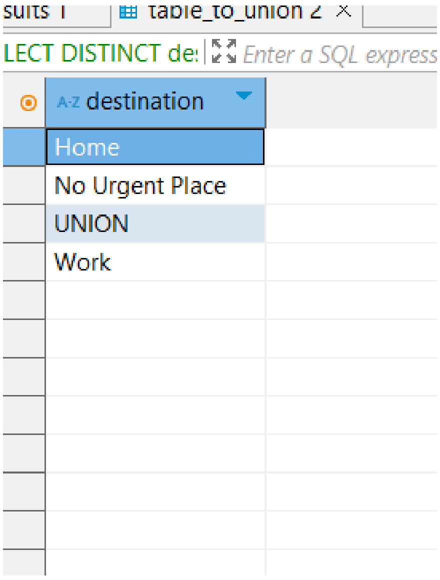


```
# SELECT occupation FROM dataset_1 GROUP BY occupation= 'Student'
df.groupby('occupation').filter(lambda x: x['occupation'].iloc[0] == 'Student').groupby('occupation').size()
```

```
occupation
Student    1584
dtype: int64
```

SQL Query

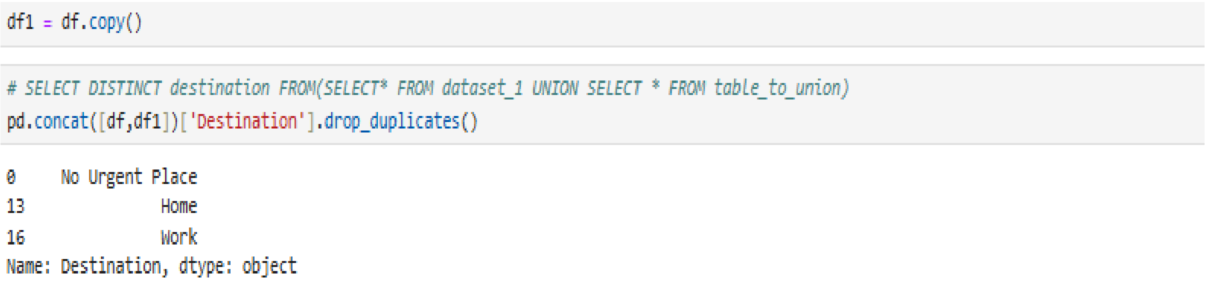
SELECT DISTINCT destination FROM (SELECT*FROM dataset_1 UNION SELECT*FROM table_to_union)



destination
Home
No Urgent Place
UNION
Work

Python Code

pd. concat ([df, df1]) ['Destination'].drop duplicates()



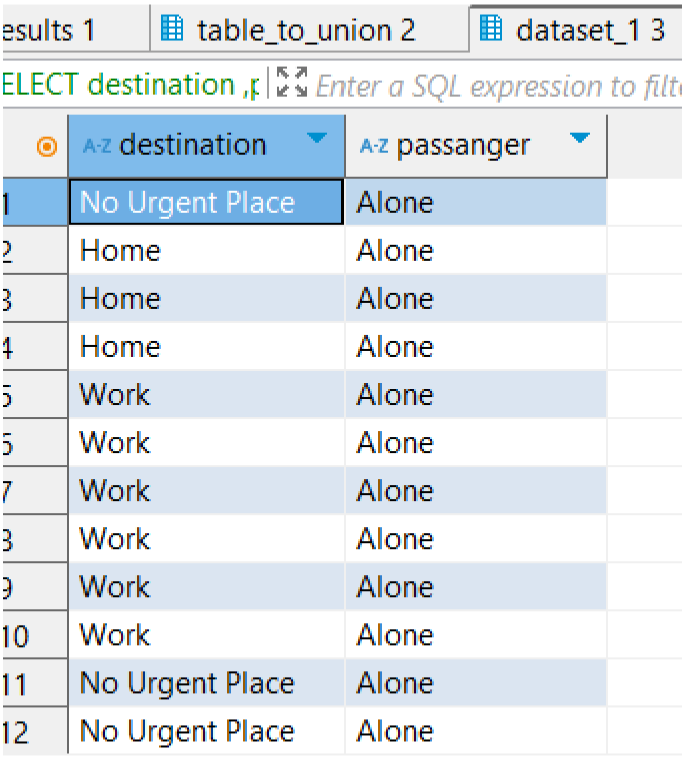
```
df1 = df.copy()
```

```
# SELECT DISTINCT destination FROM(SELECT* FROM dataset_1 UNION SELECT * FROM table_to_union)
pd.concat([df,df1])['Destination'].drop_duplicates()
```

```
0     No Urgent Place
13               Home
16               Work
Name: Destination, dtype: object
```

SQL Query

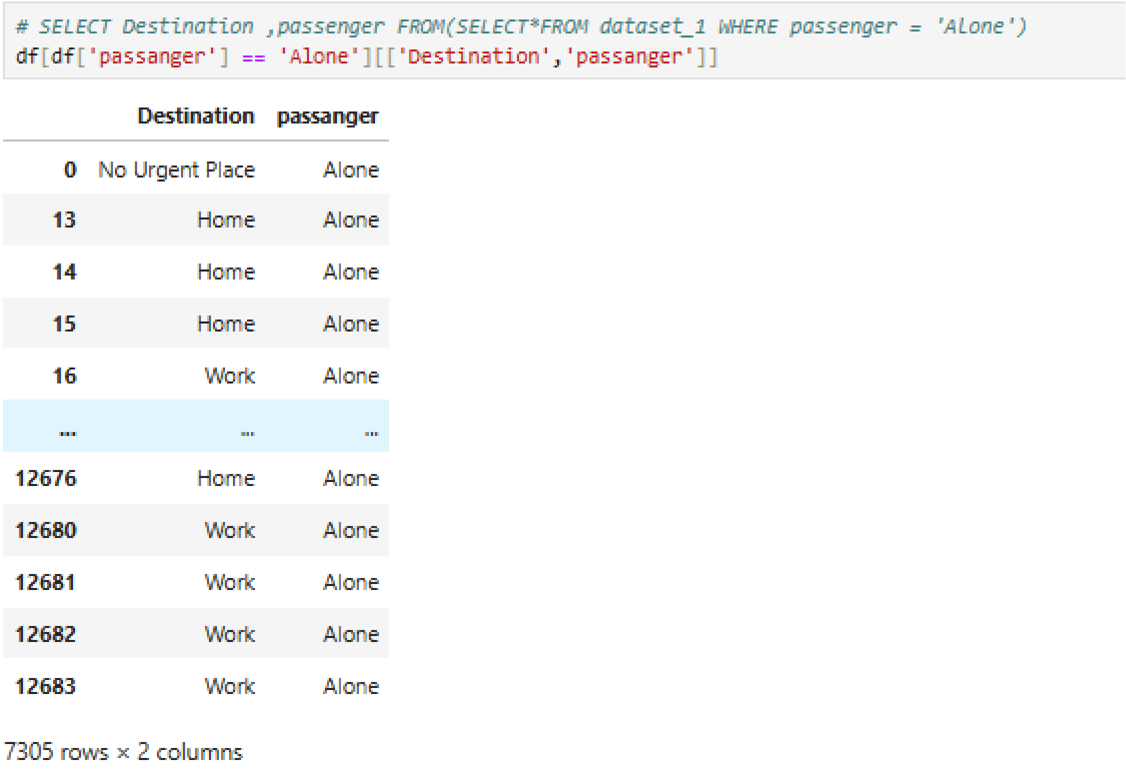
SELECT destination, passanger FROM (SELECT*FROM dataset_1 WHERE passanger = 'Alone')



	destination	passanger
1	No Urgent Place	Alone
2	Home	Alone
3	Home	Alone
4	Home	Alone
5	Work	Alone
6	Work	Alone
7	Work	Alone
8	Work	Alone
9	Work	Alone
10	Work	Alone
11	No Urgent Place	Alone
12	No Urgent Place	Alone

Python Code

df[df[‘passanger’] == ‘Alone’][[‘Destination’,’passanger’]]



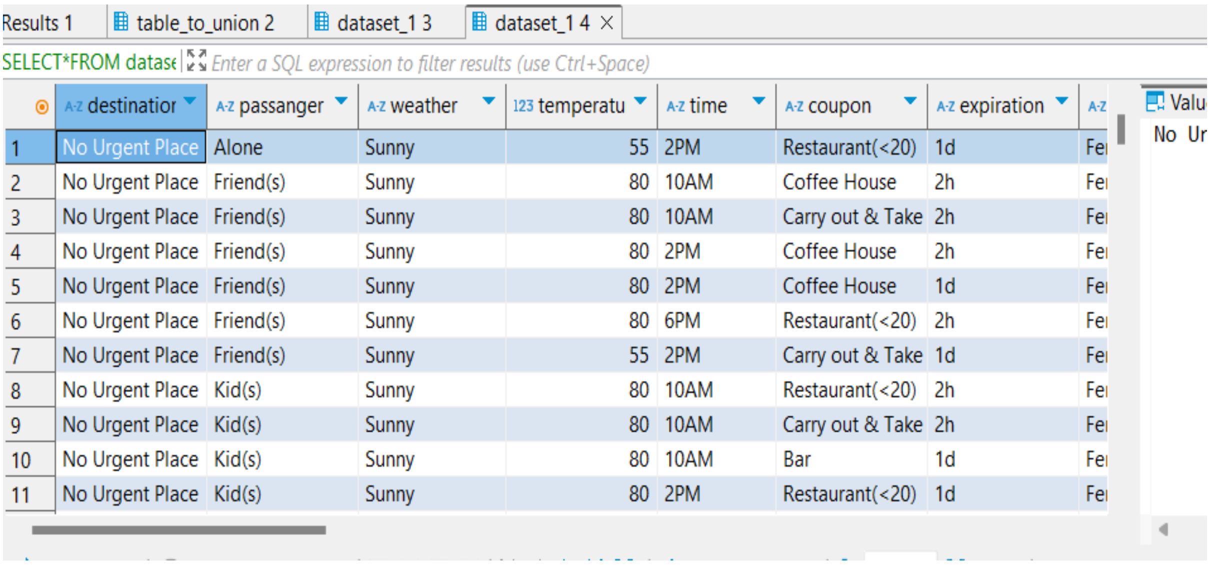
```
# SELECT Destination ,passenger FROM(SELECT*FROM dataset_1 WHERE passenger = 'Alone')
df[df['passanger'] == 'Alone'][['Destination','passanger']]
```

	Destination	passanger
0	No Urgent Place	Alone
13	Home	Alone
14	Home	Alone
15	Home	Alone
16	Work	Alone
...	...	...
12676	Home	Alone
12680	Work	Alone
12681	Work	Alone
12682	Work	Alone
12683	Work	Alone

7305 rows × 2 columns

SQL Query

SELECT*FROM dataset_1 WHERE weather LIKE ‘Sun%’



Results 1 | table_to_union 2 | dataset_1 3 | dataset_1 4

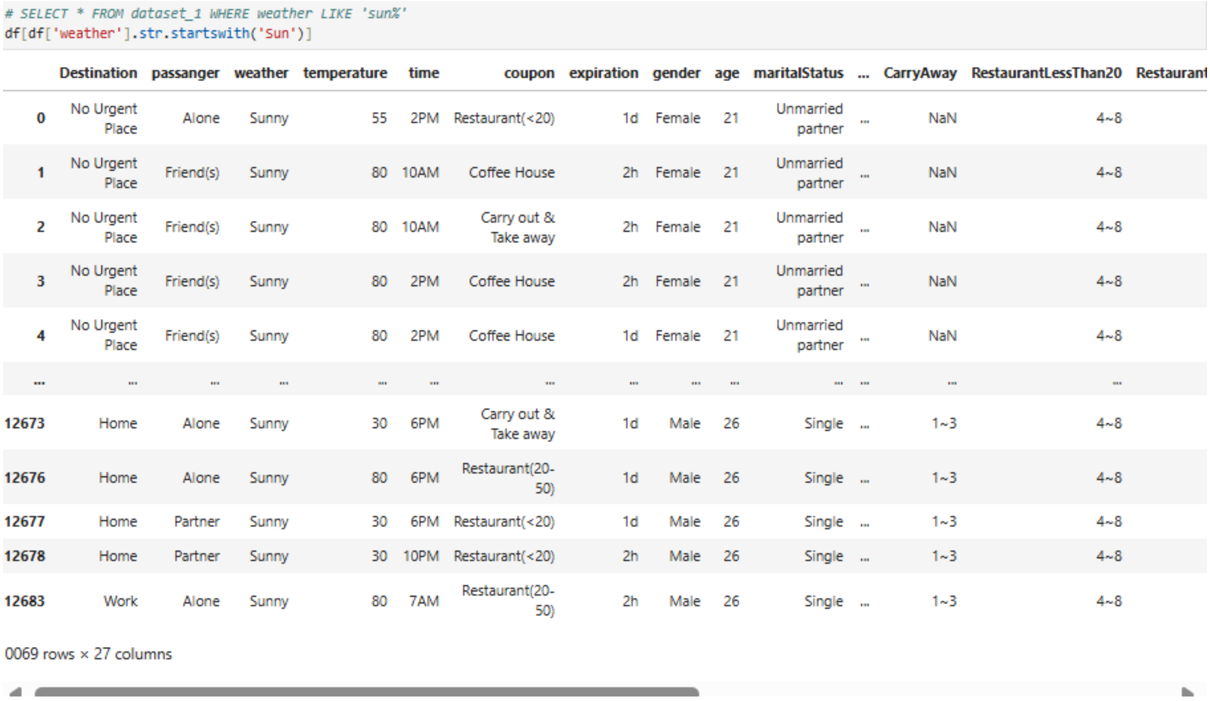
SELECT*FROM datas Enter a SQL expression to filter results (use Ctrl+Space)

	destinatior	passanger	weather	temperatu	time	coupon	expiration	
1	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1d	Fe
2	No Urgent Place	Friend(s)	Sunny	80	10AM	Coffee House	2h	Fe
3	No Urgent Place	Friend(s)	Sunny	80	10AM	Carry out & Take	2h	Fe
4	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	2h	Fe
5	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	1d	Fe
6	No Urgent Place	Friend(s)	Sunny	80	6PM	Restaurant(<20)	2h	Fe
7	No Urgent Place	Friend(s)	Sunny	55	2PM	Carry out & Take	1d	Fe
8	No Urgent Place	Kid(s)	Sunny	80	10AM	Restaurant(<20)	2h	Fe
9	No Urgent Place	Kid(s)	Sunny	80	10AM	Carry out & Take	2h	Fe
10	No Urgent Place	Kid(s)	Sunny	80	10AM	Bar	1d	Fe
11	No Urgent Place	Kid(s)	Sunny	80	2PM	Restaurant(<20)	1d	Fe

Valu
No Ur

Python Code

df[df[‘weather’]. str. startswith(‘Sun’)]



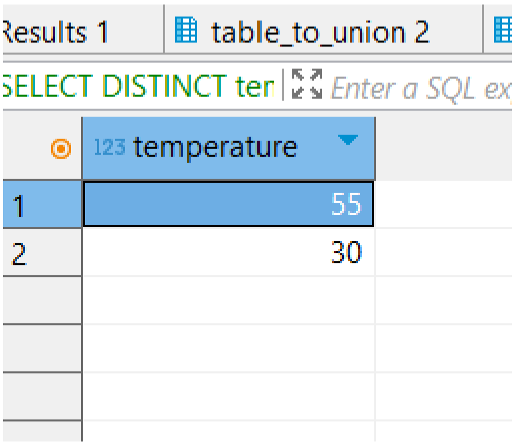
```
# SELECT * FROM dataset_1 WHERE weather LIKE 'sun%'
df[df['weather'].str.startswith('Sun')]
```

	Destination	passanger	weather	temperature	time	coupon	expiration	gender	age	maritalStatus	...	CarryAway	RestaurantLessThan20	Restaurant
0	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1d	Female	21	Unmarried partner	...	NaN	4~8	
1	No Urgent Place	Friend(s)	Sunny	80	10AM	Coffee House	2h	Female	21	Unmarried partner	...	NaN	4~8	
2	No Urgent Place	Friend(s)	Sunny	80	10AM	Carry out & Take away	2h	Female	21	Unmarried partner	...	NaN	4~8	
3	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	2h	Female	21	Unmarried partner	...	NaN	4~8	
4	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	1d	Female	21	Unmarried partner	...	NaN	4~8	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
12673	Home	Alone	Sunny	30	6PM	Carry out & Take away	1d	Male	26	Single	...	1~3	4~8	
12676	Home	Alone	Sunny	80	6PM	Restaurant(20-50)	1d	Male	26	Single	...	1~3	4~8	
12677	Home	Partner	Sunny	30	6PM	Restaurant(<20)	1d	Male	26	Single	...	1~3	4~8	
12678	Home	Partner	Sunny	30	10PM	Restaurant(<20)	2h	Male	26	Single	...	1~3	4~8	
12683	Work	Alone	Sunny	80	7AM	Restaurant(20-50)	2h	Male	26	Single	...	1~3	4~8	

0069 rows × 27 columns

SQL Query

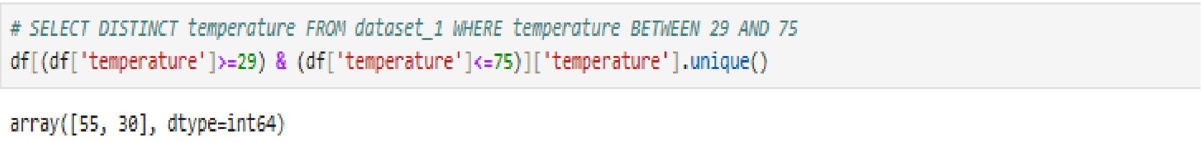
SELECT DISTINCT temperature FROM dataset_1 WHERE temperature BETWEEN 29 AND 75



	123 temperature
1	55
2	30

Python Code

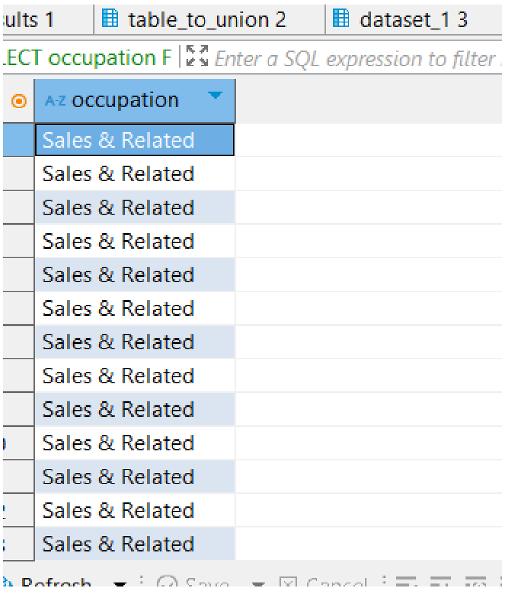
df[(df['temperature']>=29 & (df['temperature'] <=75)]['temperature']. unique ()



```
# SELECT DISTINCT temperature FROM dataset_1 WHERE temperature BETWEEN 29 AND 75
df[(df['temperature']>=29) & (df['temperature']<=75)]['temperature'].unique()

array([55, 30], dtype=int64)
```

SQL Query

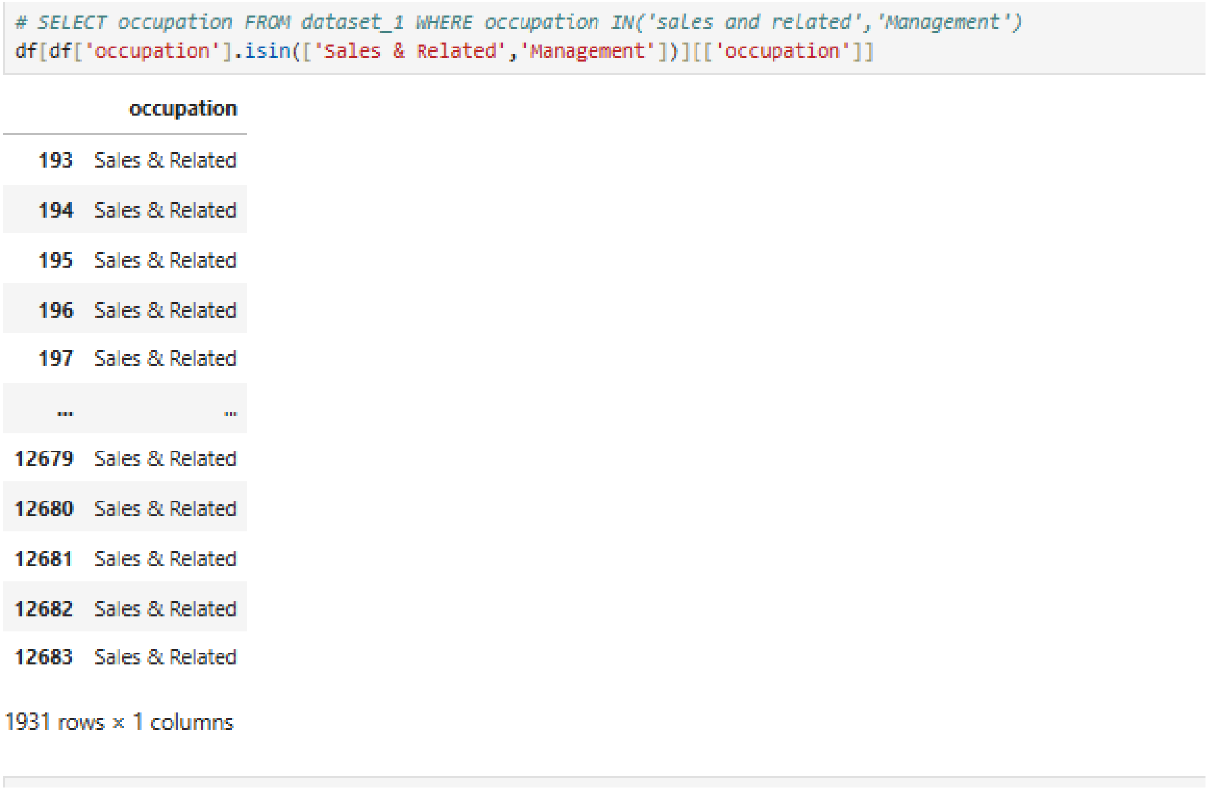
SELECT occupation FROM dataset_1 WHERE occupation IN ('Sales & Related','Management')



A-Z occupation
Sales & Related
Sales & Related
Sales & Related
Sales & Related
Sales & Related
Sales & Related
Sales & Related
Sales & Related
Sales & Related
Sales & Related
Sales & Related
Sales & Related
Sales & Related

Python Code

df[df[‘occupation’]. isin ([‘Sales & Related’, ‘Management’])][[‘occupation’]]



```
# SELECT occupation FROM dataset_1 WHERE occupation IN('sales and related','Management')
df[df['occupation'].isin(['Sales & Related','Management'])][['occupation']]
```

```
             occupation
  193   Sales & Related
  194   Sales & Related
  195   Sales & Related
  196   Sales & Related
  197   Sales & Related
  ...               ...
12679   Sales & Related
12680   Sales & Related
12681   Sales & Related
12682   Sales & Related
12683   Sales & Related

1931 rows × 1 columns
```