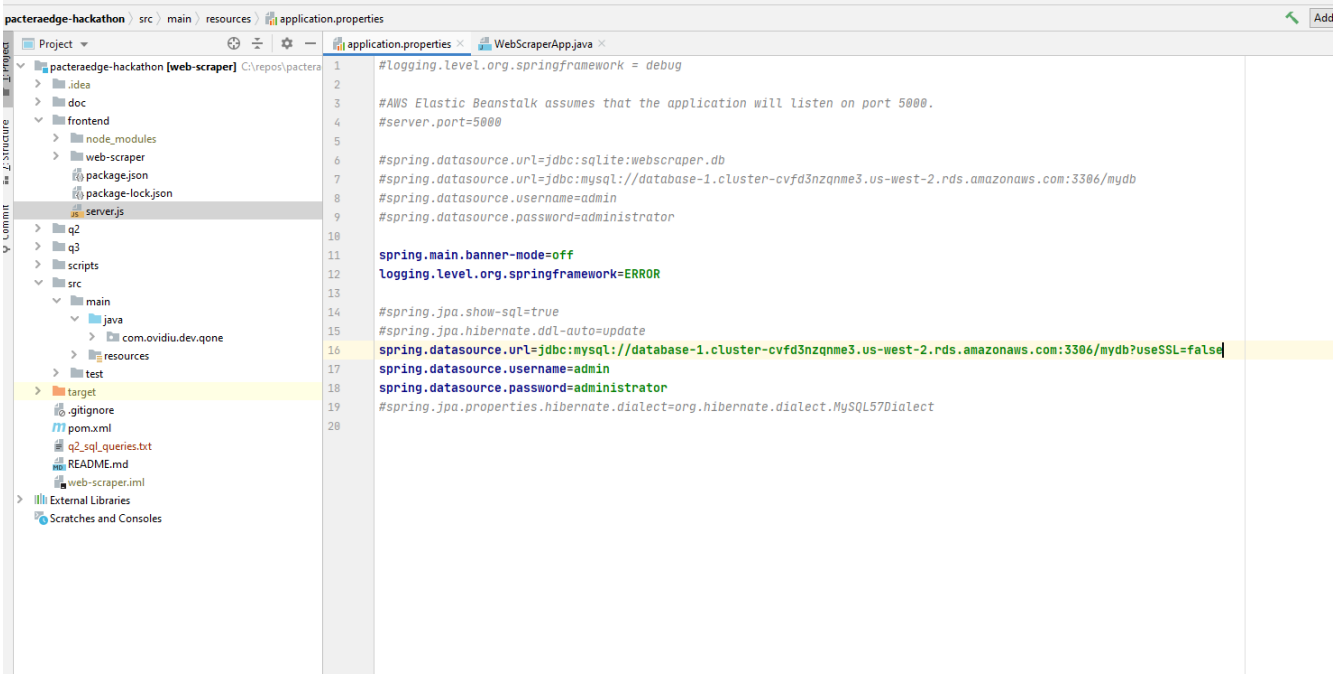
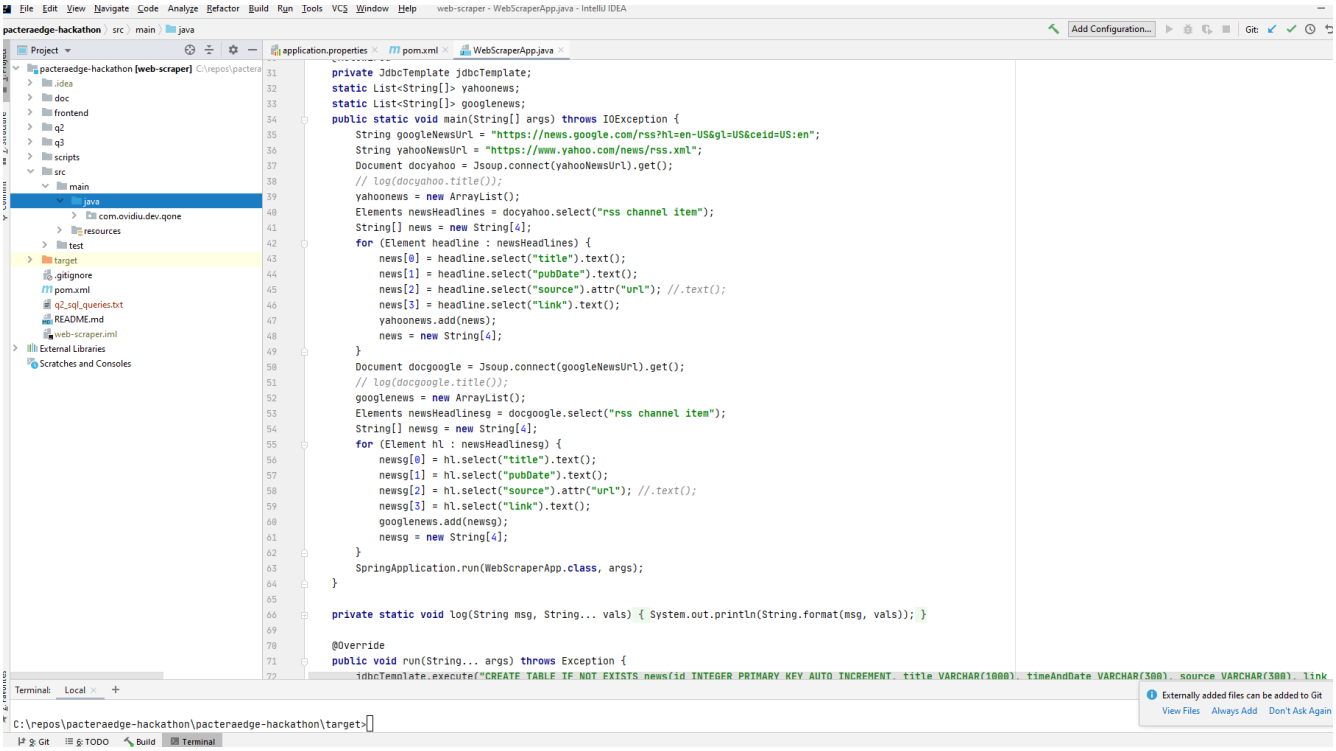
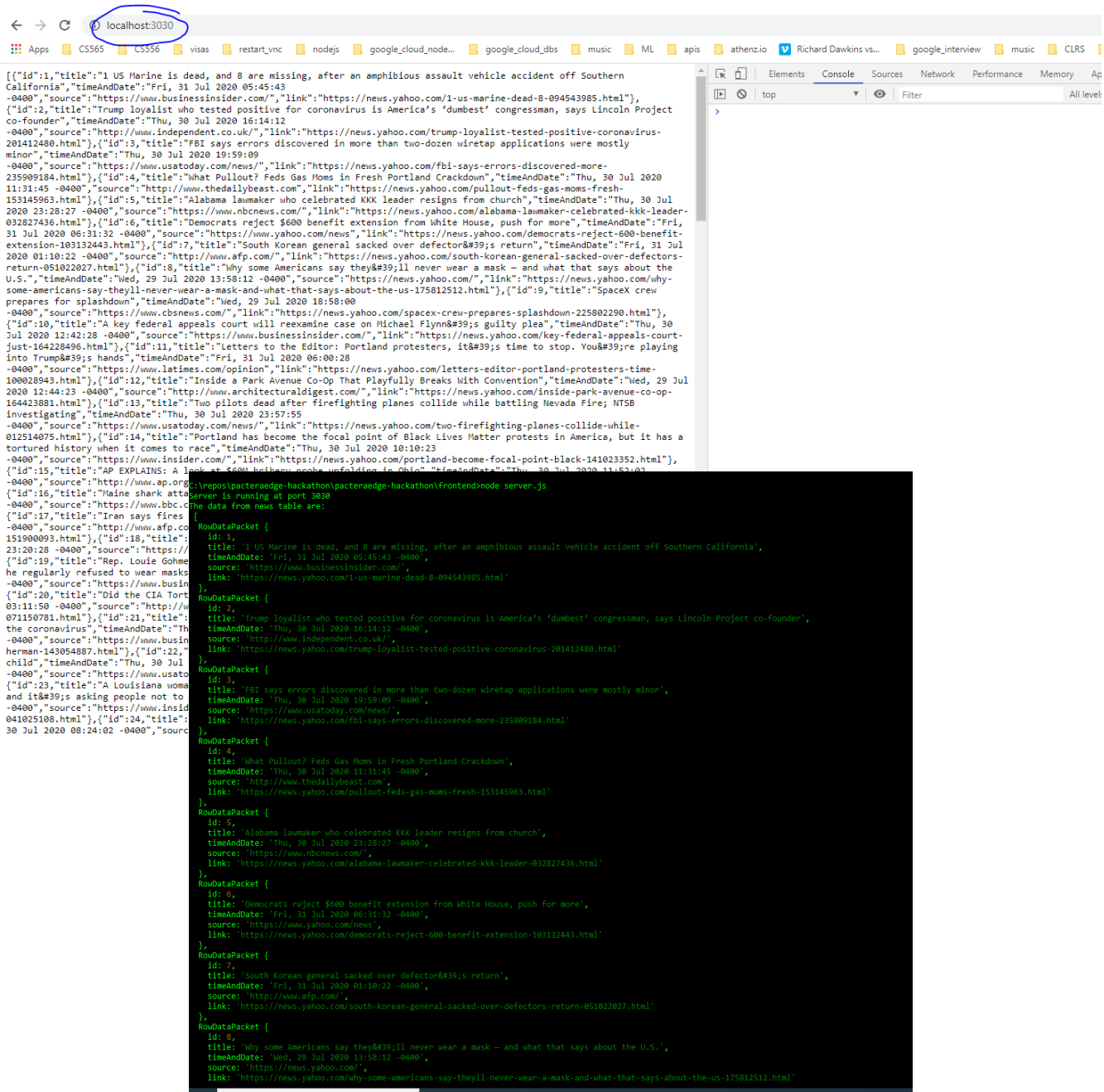
Ovidiu Mura
git repo: https://github.com/ovimura/pacteraedge-hackathon

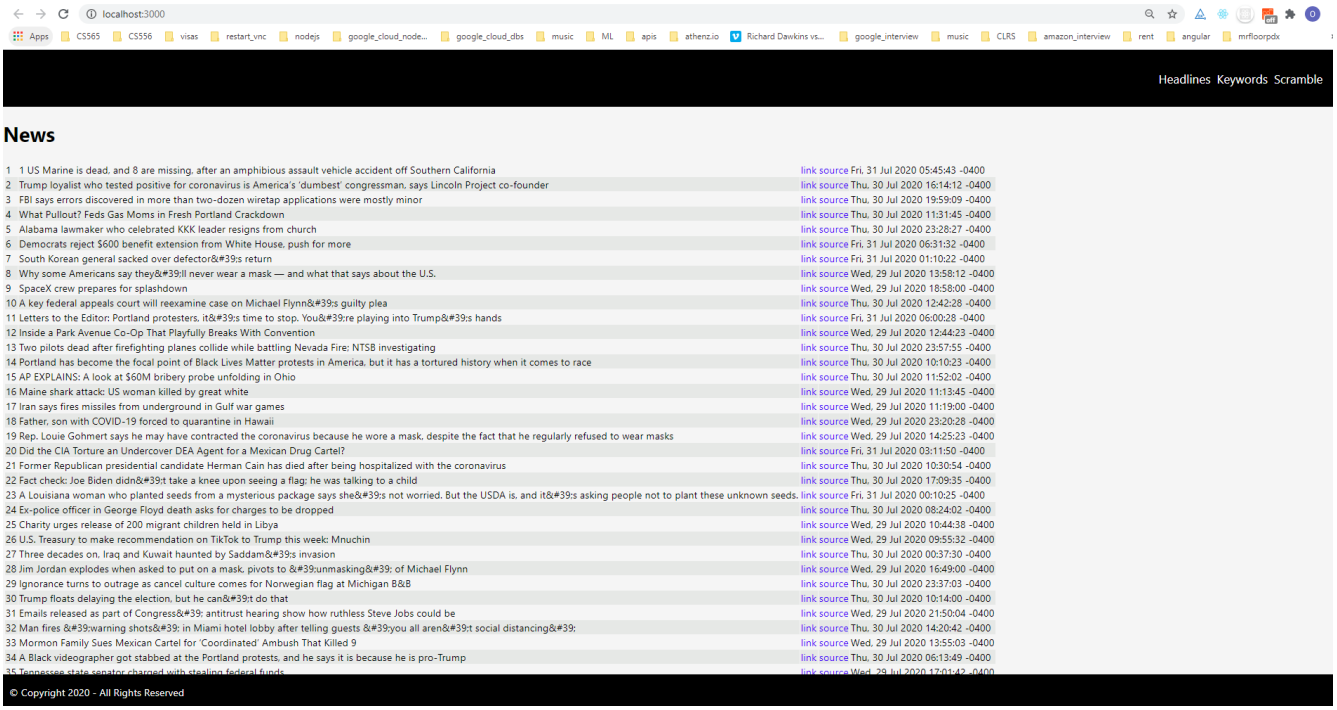
1.



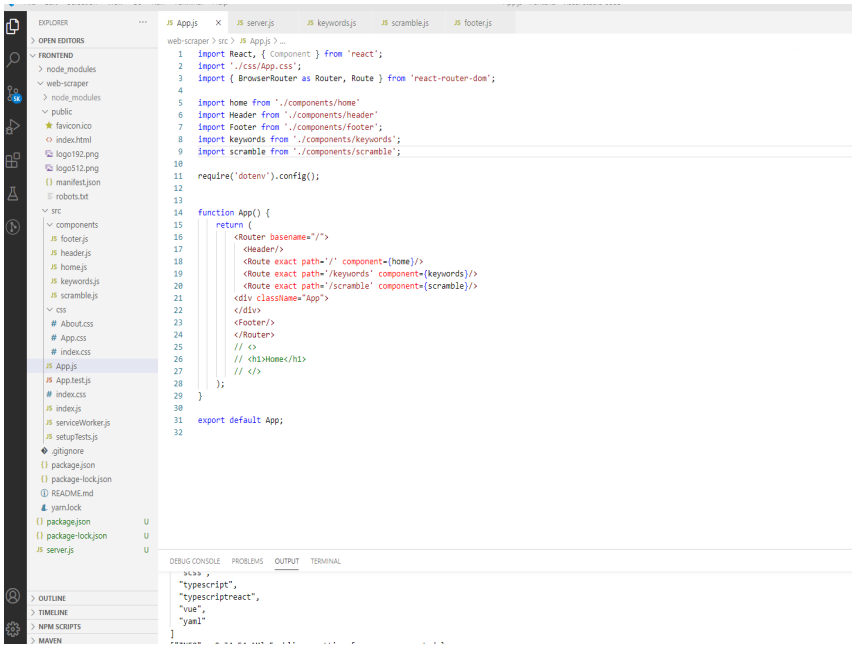


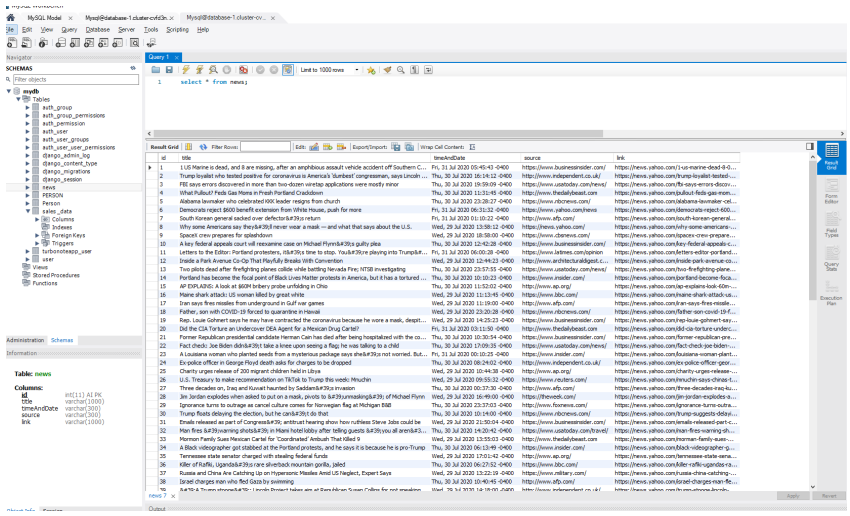
```
 Volume in drive C has no label.
 Volume Serial Number is 723F-67C1

 Directory of C:\repos\pacteraedge-hackathon\pacteraedge-hackathon\frontend

07/31/2020  07:58 AM    <DIR>          .
07/31/2020  07:58 AM    <DIR>          ..
07/31/2020  07:49 AM    <DIR>          node_modules
07/31/2020  07:46 AM            17,146 package-lock.json
07/31/2020  07:46 AM               278 package.json
07/31/2020  09:44 AM               762 server.js
07/31/2020  09:34 AM    <DIR>          web-scraper
               3 File(s)         18,186 bytes
               4 Dir(s)  407,156,961,280 bytes free

C:\repos\pacteraedge-hackathon\pacteraedge-hackathon\frontend>node server.js
Server is running at port 3030
```





git repo: https://github.com/ovimura/pacteraedge-hackathon





2.

Query 2:

For each type of the stores, find 5 stores with highest average weekly sales.

Solution:

```
select tp, avg_w from
(select st.Type tp,st.Store sto, avg(sa.Weekly_Sales) avg_w from stores_data st join sales_data sa on
(st.Store = sa.Store)
group by st.Type, st.Store
having avg_w > (select Weekly_Sales from stores_data))
group by avg_w order by 2 limit 5
```

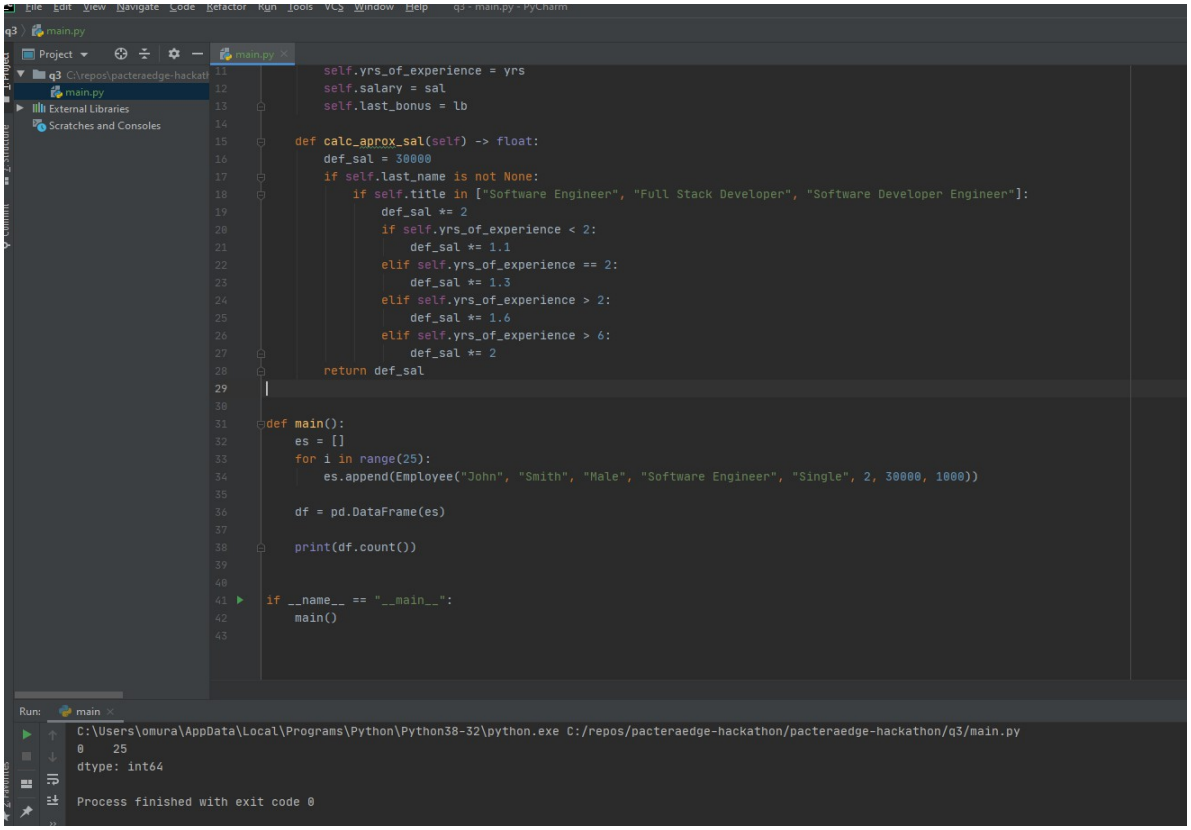
Query 3:

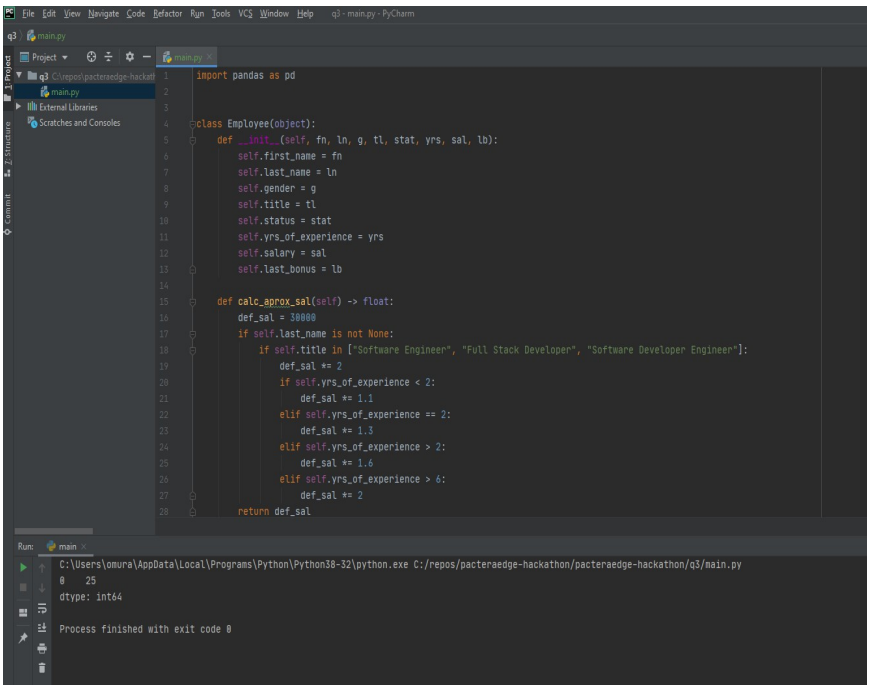
Find stores with increased total sales in December from 2010 to 2011.

```
SELECT st.Store a, sum(sd.Weekly_Sales) b
from stores_data st join sales_data sd on st.Store = sd.Store
where SUBSTR(sd.date, 7, 4) IS '2010' and SUBSTR(sd.date, 4, 2) IS '12'
group by a;

and b > 1000;
(
SELECT sum(sd1.Weekly_Sales) from stores_data st1 join sales_data sd1 on st1.Store = sd1.Store
where SUBSTR(sd1.date, 7, 4) = '2011' and SUBSTR(sd1.date, 4, 2) = '12'
```

and a = st1.Store
);

3.





```
import pandas as pd
class Employee(object):
	def __init__(self, fn, ln, g, tl, stat, yrs, sal, lb):
		self.first_name = fn
		self.last_name = ln
		self.gender = g
		self.title = tl
		self.status = stat
		self.yrs_of_experience = yrs
		self.salary = sal
		self.last_bonus = lb
	def calc_aprox_sal(self) -> float:
		def_sal = 30000
		if self.last_name is not None:
			if self.title in ['Software Engineer', 'Full Stack Developer', 'Software Developer Engineer']:
				def_sal *= 2
				if self.yrs_of_experience < 2:
					def_sal *= 1.1
				elif self.yrs_of_experience == 2:
					def_sal *= 1.3
				elif self.yrs_of_experience > 2:
					def_sal *= 1.6
				elif self.yrs_of_experience > 6:
					def_sal *= 2
		return def_sal
def main():
	es = []
	for i in range(25):
		es.append(Employee('John', 'Smith', 'Male', 'Software Engineer', 'Single', 2, 30000, 1000))
	df = pd.DataFrame(es)
	print(df.count())
if __name__ == '__main__':
	main()
```