Average Salary: <https://platform.stratascratch.com/coding/9917-average-salaries?code_type=2>

import pandas as pd

# Start writing code

df=employee.groupby('department')['salary'].mean().reset\_index()

df=df.rename(columns={'salary':'avg\_salary'})

merge\_data=pd.merge(employee,df,on='department',how='left')

merge\_data[['department','first\_name','last\_name','salary','avg\_salary']]

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Customer Details:<https://platform.stratascratch.com/coding/9891-customer-details?code_type=3>

select c.first\_name,c.last\_name,c.city,o.order\_details from customers c

left join orders o

on c.id=o.cust\_id

order by 1

import pandas as pd

# Start writing code

merged\_data=pd.merge(customers,orders,left\_on='id',right\_on='cust\_id',how='left')

final=merged\_data[['first\_name','last\_name','city','order\_details']]

final.sort\_values(by=['first\_name'])

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Updated Records:<https://platform.stratascratch.com/coding/10299-finding-updated-records?code_type=5>

import pandas as pd

# Start writing code

df=ms\_employee\_salary.sort\_values(by=['id','salary'],ascending=[True,False]).groupby('id').first().reset\_index()

df[['id','first\_name','last\_name','salary','department\_id']]

with cte as

(

select \*,rank() over(partition by id order by salary desc ) as rn from ms\_employee\_salary

)

select id,first\_name,last\_name,salary,department\_id

from cte

where rn=1

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Bathrooms and Bedrooms:<https://platform.stratascratch.com/coding/9622-number-of-bathrooms-and-bedrooms?code_type=3>

select city,property\_type,avg(bathrooms) as n\_bathrooms\_avg,avg(bedrooms) as n\_bedrooms\_avg from airbnb\_search\_details

group by city,property\_type

import pandas as pd

# Start writing code

airbnb\_search\_details.groupby(['city','property\_type'])[['bedrooms','bathrooms']].mean().reset\_index().rename(columns={'bedrooms':'avg bedrooms','bathrooms':'avg bathroom'})

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Order Details**

<https://platform.stratascratch.com/coding/9913-order-details?code_type=2>

import pandas as pd

# Start writing code

df=pd.merge(orders,customers,left\_on='cust\_id',right\_on='id',how='left')

df[df['first\_name'].isin(['Jill','Eva'])].sort\_values(by='cust\_id')[['first\_name','order\_date','order\_details','total\_order\_cost']]

select first\_name,order\_date,order\_details,Total\_order\_cost from orders o

left join customers c

on o.cust\_id=c.id

where c.first\_name in('Jill','Eva')

order by cust\_id

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

# Salaries Differences

<https://platform.stratascratch.com/coding/10308-salaries-differences?code_type=3>

with cte as

(

select department,max(salary) as max\_sal from db\_employee a

join db\_dept b

on a.department\_id=b.id

where b.department in('marketing','engineering')

group by department

)

select abs(max\_sal-max\_sal\_2) as salary\_difference

from

(

select max\_sal,lead(max\_sal) over() as max\_sal\_2 from cte) w

where w.max\_sal-2 is not null

df=pd.merge(db\_employee,db\_dept,left\_on='department\_id',right\_on='id',how='left')

abs(df[df['department'].isin(['marketing','engineering'])].groupby('department')['salary'].max().reset\_index()['salary'].diff().iloc[1])

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

# Popularity of Hack

https://platform.stratascratch.com/coding/10061-popularity-of-hack?code\_type=3

select location,avg(popularity) as avg\_popularity from facebook\_hack\_survey a

join facebook\_employees b

on a.employee\_id=b.id

group by location

order by 1

import pandas as pd

# Start writing code

df=pd.merge(facebook\_employees,facebook\_hack\_survey,left\_on='id',right\_on='employee\_id',how='left')

df.groupby('location')['popularity'].mean().reset\_index()

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

# Count the number of user events performed by MacBookPro users

<https://platform.stratascratch.com/coding/9653-count-the-number-of-user-events-performed-by-macbookpro-users?code_type=3>

select event\_name,count(\*) as event\_count from playbook\_events

where device in('macbook pro')

group by event\_name

order by 2 desc

# Import your libraries

import pandas as pd

# Start writing code

playbook\_events[playbook\_events['device']=='macbook pro'].groupby('event\_name')['user\_id'].count().reset\_index().sort\_values(by='user\_id',ascending=False).rename(columns={'user\_id':'event\_count'})

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

<https://platform.stratascratch.com/coding/10087-find-all-posts-which-were-reacted-to-with-a-heart?code_type=3>

# Find all posts which were reacted to with a heart

select distinct a.\* from facebook\_posts a

left join facebook\_reactions b

on a.post\_id=b.post\_id

where b.reaction='heart'

# Import your libraries

import pandas as pd

# Start writing code

df=pd.merge(facebook\_posts,facebook\_reactions,on='post\_id',how='left')

df[df['reaction']=='heart'][['post\_id','poster\_x','post\_text','post\_keywords','post\_date']].rename(columns={'poster\_x':'poster'}).drop\_duplicates()

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

# Lyft Driver Wages

<https://platform.stratascratch.com/coding/10003-lyft-driver-wages?code_type=3>

select \* from lyft\_drivers

where yearly\_salary<=30000 or yearly\_salary>=70000

# Import your libraries

import pandas as pd

# Start writing code

lyft\_drivers[(lyft\_drivers['yearly\_salary']<=30000) |(lyft\_drivers['yearly\_salary']>=70000)].sort\_values(by='index')

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

# Top Ranked Songs

<https://platform.stratascratch.com/coding/9991-top-ranked-songs?code_type=3>

select trackname,count(\*) as times\_top1 from spotify\_worldwide\_daily\_song\_ranking

where position=1

group by trackname

order by 2 desc

import pandas as pd

# Start writing code

spotify\_worldwide\_daily\_song\_ranking[spotify\_worldwide\_daily\_song\_ranking['position']==1].groupby('trackname')['id'].count().reset\_index().sort\_values(by='id',ascending=False).rename(columns={'id':'times\_top1'})

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

<https://platform.stratascratch.com/coding/10176-bikes-last-used?code_type=3>

# Bikes Last Used

select bike\_number,max(end\_time) from dc\_bikeshare\_q1\_2012

group by bike\_number

# Import your libraries

import pandas as pd

# Start writing code

dc\_bikeshare\_q1\_2012.groupby('bike\_number')['end\_time'].max().reset\_index().rename(columns={'end\_time':'last\_used'})

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++