In [2]:
```
import pandas as pd
import numpy as np
```

In [3]:
```
twitch = {'user_id': {0: 0, 1: 2, 2: 3, 3: 1, 4: 2, 5: 0, 6: 0, 7: 3, 8: 1, 9: 2}, 'session_start': {0: '2020-08-11 0
```

In [5]:
```
df_twitch = pd.DataFrame(twitch)
df_twitch
```

Out[5]:

	user_id	session_start	session_end	session_id	session_type
0	0	2020-08-11 05:51:31	2020-08-11 05:54:45	539	streamer
1	2	2020-07-11 03:36:54	2020-07-11 03:37:08	840	streamer
2	3	2020-11-26 11:41:47	2020-11-26 11:52:01	848	streamer
3	1	2020-11-19 06:24:24	2020-11-19 07:24:38	515	viewer
4	2	2020-11-14 03:36:05	2020-11-14 03:39:19	646	viewer
5	0	2020-03-11 03:01:40	2020-03-11 03:01:59	782	streamer
6	0	2020-08-11 03:50:45	2020-08-11 03:55:59	815	viewer
7	3	2020-10-11 22:15:14	2020-10-11 22:18:28	630	viewer
8	1	2020-11-20 06:59:57	2020-11-20 07:20:11	907	streamer
9	2	2020-07-11 14:32:19	2020-07-11 14:42:33	949	viewer

In [14]:
```
df_twitch['is_streamer'] = np.where(df_twitch['session_type'] == "streamer",1,0)
```

In [15]:
```
df_twitch['is_viewer'] = np.where(df_twitch['session_type'] == "viewer",1,0)
```

In [35]:
```
ash = df_twitch[["user_id","is_streamer","is_viewer"]]
```

In [40]:
```
streamer = ash.groupby('is_streamer')['user_id'].apply(list).reset_index().iloc[1,1]
```

In [41]:
```
streamer
```

Out[41]:
```
[0, 2, 3, 0, 1]
```

In [42]:
```
viewer = ash.groupby('is_viewer')['user_id'].apply(list).reset_index().iloc[1,1]
```

In [43]:
```
set(streamer).intersection(set(viewer))
```

Out[43]:
```
{0, 1, 2, 3}
```

Question workers

In [44]:
```
workers = {'worker_id': {0: 1, 1: 2, 2: 3, 3: 4, 4: 5, 5: 6, 6: 7, 7: 8}, 'first_name': {0: 'Monika', 1: 'Niharika',
```

In [45]:
```
df_workers = pd.DataFrame(workers)
```

In [46]:
```
df_workers
```

Out[46]:

	worker_id	first_name	last_name	salary	joining_date	department
0	1	Monika	Arora	100000	2014-02-20 09:00:00	HR
1	2	Niharika	Verma	80000	2014-06-11 09:00:00	Admin
2	3	Vishal	Singhal	300000	2014-02-20 09:00:00	HR
3	4	Amitah	Singh	500000	2014-02-20 09:00:00	Admin
4	5	Vivek	Bhati	500000	2014-06-11 09:00:00	Admin
5	6	Vipul	Diwan	200000	2014-06-11 09:00:00	Account
6	7	Satish	Kumar	75000	2014-01-20 09:00:00	Account
7	8	Geetika	Chauhan	90000	2014-04-11 09:00:00	Admin

In [48]:
```
df_workers.sort["salary"],ascending = False
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [48], in <cell line: 1>()
----> 1 df_workers[df_workers["salary"],ascending == False]

NameError: name 'ascending' is not defined
```

In [ ]:

In [ ]:

Question 2

In [49]:
```
facebook_user_interactions = {'day': {0: 0, 1: 0, 2: 0, 3: 0, 4: 1, 5: 1, 6: 1, 7: 2, 8: 2, 9: 2}, 'user1': {0: 0, 1:
```

In [51]:
```
df_facebook_user_interactions = pd.DataFrame(facebook_user_interactions)
```

In [52]:
```
df_facebook_user_interactions
```

Out[52]:

	day	user1	user2
0	0	0	1
1	0	1	0
2	0	2	1
3	0	2	3
4	1	2	0
5	1	2	1
6	1	2	3
7	2	4	1
8	2	4	2
9	2	4	0

In [59]:
```
df_facebook_user_interactions1 = df_facebook_user_interactions[df_facebook_user_interactions["day"].isin([0,2])]
```

In [60]:
```
df_facebook_user_interactions1["total_interactions"] = df_facebook_user_interactions1["user1"] + df_facebook_user_int
```

```
C:\Users\mause\AppData\Local\Temp\ipykernel_4396\2686500071.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_facebook_user_interactions1["total_interactions"] = df_facebook_user_interactions1["user1"] + df_facebook_user_interactions1["user2"]
```

In [61]:
```
df_facebook_user_interactions1
```

Out[61]:

	day	user1	user2	total_interactions
0	0	0	1	1
1	0	1	0	1
2	0	2	1	3
3	0	2	3	5
7	2	4	1	5
8	2	4	2	6
9	2	4	0	4

In [68]:
```
df_facebook_user_interactions[df_facebook_user_interactions["day"].isin([0,2])].groupby("day")["user1"].count().to_cs
```

In [ ]:
```

```

Question 3

In [72]:
```
facebook_product_features_realizations = {'feature_id': {0: 0, 1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 0, 9: 0,
```

In [73]:
```
facebook_product_features_realizations = pd.DataFrame(facebook_product_features_realizations)
```

In [74]:
```
facebook_product_features_realizations
```

Out[74]:

	feature_id	user_id	step_reached	timestamp
0	0	0	1	2019-03-11 17:15:00
1	0	0	2	2019-03-11 17:22:00
2	0	0	3	2019-03-11 17:25:00
3	0	0	4	2019-03-11 17:27:00
4	0	1	1	2019-03-11 19:51:00
5	0	1	2	2019-03-11 19:52:00
6	0	1	3	2019-03-11 19:55:00
7	0	1	4	2019-03-11 19:57:00
8	0	1	5	2019-03-11 19:59:00
9	0	2	1	2019-03-11 10:01:00
10	0	2	2	2019-03-11 10:04:00
11	0	2	3	2019-03-11 10:09:00
12	1	3	6	2019-04-05 08:08:08
13	1	2	7	2019-04-05 11:12:13
14	1	1	3	2019-04-05 13:00:07

In [76]:
```
facebook_product_features_realizations.groupby("feature_id")["step_reached"].max().to_csv(header = None, sep = "|")
```

Out[76]:
```
'0|5\r\n1|7\r\n'
```

In [ ]:

In [ ]:

question 5

In [77]:
```
forbes_global_2010_2014 = {'company': {0: 'ICBC', 1: 'China Construction Bank', 2: 'Agricultural Bank of China', 3:
```

In [78]:
```
forbes_global_2010_2014 = pd.DataFrame(forbes_global_2010_2014)
```

In [79]:
```
forbes_global_2010_2014
```

Out[79]:

	company	sector	industry	continent	country	marketvalue	sales	profits	assets	rank	forbeswebpage
0	ICBC	Financials	Major Banks	Asia	China	215.6	148.7	42.7	3124.9	1	http://www.forbes.com/companies/icbc/
1	China Construction Bank	Financials	Regional Banks	Asia	China	174.4	121.3	34.2	2449.5	2	http://www.forbes.com/companies/china-construc...
2	Agricultural Bank of China	Financials	Regional Banks	Asia	China	141.1	136.4	27.0	2405.4	3	http://www.forbes.com/companies/agricultural-b...
3	JPMorgan Chase	Financials	Major Banks	North America	United States	229.7	105.7	17.3	2435.3	4	http://www.forbes.com/companies/jpmorgan-chase/
4	Berkshire Hathaway	Financials	Investment Services	North America	United States	309.1	178.8	19.5	493.4	5	http://www.forbes.com/companies/berkshire-hath...
...	...	...	...	...	...	...	...	...	...	...	...
95	CVS Caremark	Consumer Staples	Drug Retail	North America	United States	87.8	126.8	4.6	71.5	96	http://www.forbes.com/companies/cvs-caremark/
96	ING Group	Financials	Life & Health Insurance	Europe	Netherlands	56.1	34.5	4.4	1488.7	97	http://www.forbes.com/companies/ing-group/
97	Saudi Basic Industries	Materials	Diversified Chemicals	Asia	Saudi Arabia	94.4	50.4	6.7	90.4	98	http://www.forbes.com/companies/saudi-basic-in...
98	Merck & Co	Health Care	Pharmaceuticals	North America	United States	165.8	44.1	4.4	105.6	99	http://www.forbes.com/companies/merck-co/
99	Walt Disney	Consumer Discretionary	Broadcasting & Cable	North America	United States	142.9	46.0	6.6	83.2	100	http://www.forbes.com/companies/walt-disney/

100 rows × 11 columns

In [80]:
```
forbes_global_2010_2014.groupby("sector")["marketvalue"].max().to_csv(header = None, sep = "|")
```

Out[80]:
```
'Consumer Discretionary|247.9\r\nConsumer Staples|239.6\r\nEnergy|422.3\r\nFinancials|309.1\r\nHealth Care|277.0\r\n
Industrials|259.6\r\nInformation Technology|483.1\r\nMaterials|182.3\r\nTelecommunication Services|197.7\r\nUtilitie
s|75.8\r\n'
```

In [83]:
```
forbes_global_2010_2014
```

Out[83]:

	company	sector	industry	continent	country	marketvalue	sales	profits	assets	rank	forbeswebpage
0	ICBC	Financials	Major Banks	Asia	China	215.6	148.7	42.7	3124.9	1	http://www.forbes.com/companies/icbc/
1	China Construction Bank	Financials	Regional Banks	Asia	China	174.4	121.3	34.2	2449.5	2	http://www.forbes.com/companies/china-construc...
2	Agricultural Bank of China	Financials	Regional Banks	Asia	China	141.1	136.4	27.0	2405.4	3	http://www.forbes.com/companies/agricultural-b...
3	JPMorgan Chase	Financials	Major Banks	North America	United States	229.7	105.7	17.3	2435.3	4	http://www.forbes.com/companies/jpmorgan-chase/
4	Berkshire Hathaway	Financials	Investment Services	North America	United States	309.1	178.8	19.5	493.4	5	http://www.forbes.com/companies/berkshire-hath...
...	...	...	...	...	...	...	...	...	...	...	...
95	CVS Caremark	Consumer Staples	Drug Retail	North America	United States	87.8	126.8	4.6	71.5	96	http://www.forbes.com/companies/cvs-caremark/
96	ING Group	Financials	Life & Health Insurance	Europe	Netherlands	56.1	34.5	4.4	1488.7	97	http://www.forbes.com/companies/ing-group/
97	Saudi Basic Industries	Materials	Diversified Chemicals	Asia	Saudi Arabia	94.4	50.4	6.7	90.4	98	http://www.forbes.com/companies/saudi-basic-in...
98	Merck & Co	Health Care	Pharmaceuticals	North America	United States	165.8	44.1	4.4	105.6	99	http://www.forbes.com/companies/merck-co/
99	Walt Disney	Consumer Discretionary	Broadcasting & Cable	North America	United States	142.9	46.0	6.6	83.2	100	http://www.forbes.com/companies/walt-disney/

100 rows × 11 columns

In [84]:
```
forbes_global_2010_2014.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 100 entries, 0 to 99
Data columns (total 11 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   company        100 non-null    object 
 1   sector         100 non-null    object 
 2   industry       100 non-null    object 
 3   continent      100 non-null    object 
 4   country        100 non-null    object 
 5   marketvalue    100 non-null    float64
 6   sales          100 non-null    float64
 7   profits        100 non-null    float64
 8   assets         100 non-null    float64
 9   rank           100 non-null    int64  
 10  forbeswebpage  100 non-null    object 
dtypes: float64(4), int64(1), object(6)
memory usage: 9.4+ KB
```

In [86]:
```
forbes_global_2010_2014.isna()
```

Out[86]:

	company	sector	industry	continent	country	marketvalue	sales	profits	assets	rank	forbeswebpage
0	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...
95	False	False	False	False	False	False	False	False	False	False	False
96	False	False	False	False	False	False	False	False	False	False	False
97	False	False	False	False	False	False	False	False	False	False	False
98	False	False	False	False	False	False	False	False	False	False	False
99	False	False	False	False	False	False	False	False	False	False	False

100 rows × 11 columns

In [87]:
```
forbes_global_2010_2014.isna().sum()
```

Out[87]:
```
company          0
sector           0
industry         0
continent        0
country          0
marketvalue      0
sales            0
profits          0
assets           0
rank             0
forbeswebpage    0
dtype: int64
```

In [90]:
```
forbes_global_2010_2014[forbes_global_2010_2014["sector"] == "Information Technology"].groupby("country")["company"].
```

Out[90]:
```
'United States|8\r\nSouth Korea|1\r\n'
```

In [ ]:
```
```

In [93]:
```
from numpy import nan
```

Question

In [94]:
```
dc_1 = {'flight_date': {0: '2015-01-02 00:00:00', 1: '2015-01-02 00:00:00', 2: '2015-01-09 00:00:00', 3: '2015-01-05
```

In [95]:
```
dc_1 = pd.DataFrame(dc_1)
```

In [96]:
```
dc_1
```

Out[96]:

	flight_date	unique_carrier	flight_num	origin	dest	arr_delay	cancelled	distance	carier_delay	weather_delay	late_aircraft_delay	na
0	2015-01-02 00:00:00	VX	231	ORD	LAX	33.0	0	1744	33.0	0.0	0.0	
1	2015-01-02 00:00:00	EV	4736	CLE	EWR	-29.0	0	404	NaN	NaN	NaN	
2	2015-01-09 00:00:00	US	2195	LGA	DCA	-7.0	0	214	NaN	NaN	NaN	
3	2015-01-05 00:00:00	EV	5586	ATL	FAY	-9.0	0	331	NaN	NaN	NaN	
4	2015-01-02 00:00:00	B6	1022	PBI	BOS	-23.0	0	1197	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	...
95	2015-01-08 00:00:00	AA	1207	PHL	DFW	NaN	1	1303	NaN	NaN	NaN	
96	2015-01-09 00:00:00	AA	1650	RNO	ORD	NaN	1	1671	NaN	NaN	NaN	
97	2015-01-05 00:00:00	AA	1201	ORD	AUS	NaN	1	978	NaN	NaN	NaN	
98	2015-01-05 00:00:00	AA	1201	AUS	ORD	NaN	1	978	NaN	NaN	NaN	
99	2015-01-11 00:00:00	AA	272	SFO	MIA	NaN	1	2585	NaN	NaN	NaN	

100 rows × 14 columns

In [98]:
```
dc_1[['origin','dest','distance']].sort_values(by='distance',ascending=False).head()
```

Out[98]:

	origin	dest	distance
99	SFO	MIA	2585
8	SFO	MIA	2585
6	JFK	SMF	2521
75	KOA	LAX	2504
14	OGG	SFO	2338

In [ ]: