In [1]:

```
import pandas as pd
```

Будем работать с датасетом по оттоку клиентов из банка https://www.kaggle.com/datasets/shubh0799/churn-modelling, но датасет из себя будет представлять две таблицы:

1. Личные данные клиента

    A. CustomerId - Уникальный идентификатор клиента
    B. Surname - Фамилия клиента
    C. Geography - Из какой страны клиент
    D. Gender - Пол клиента
    E. Age - Возраст клиента
    F. EstimatedSalary - Предположительная зарплата клиента

2. Данные по поведению клиента в банке

    A. CustomerId - Уникальный идентификатор клиента
    B. CustomerId - Уникальный идентификатор клиента
    C. Tenure - Сколько лет человек является клиентом банка
    D. Balance - Баланс счета
    E. NumOfProducts - Количество открытых продуктов
    F. HasCrCard - Есть ли у клиента кредитная карта
    G. IsActiveMember - Является ли клиент активные участником
    H. Exited - Уйдет ли человек в отток

In [2]:

```
users = pd.read_csv('users.csv', sep=';')
users.head()
```

Out[2]:

	CustomerId	Surname	Geography	Gender	Age	EstimatedSalary
0	15634602	Hargrave	France	Female	42	101348.88
1	15647311	Hill	Spain	Female	41	112542.58
2	15619304	Onio	France	Female	42	113931.57
3	15701354	Boni	France	Female	39	93826.63
4	15737888	Mitchell	Spain	Female	43	79084.10

In [3]:

```
users.shape
```

Out[3]:

```
(9998, 6)
```

Создание новых признаков

In [4]:

```
users['new_feature'] = 0
users.head()
```

Out[4]:

	CustomerId	Surname	Geography	Gender	Age	EstimatedSalary	new_feature
0	15634602	Hargrave	France	Female	42	101348.88	0
1	15647311	Hill	Spain	Female	41	112542.58	0
2	15619304	Onio	France	Female	42	113931.57	0
3	15701354	Boni	France	Female	39	93826.63	0
4	15737888	Mitchell	Spain	Female	43	79084.10	0

In [5]:

```
users['Age (days)'] = users['Age'] * 365
users.head()
```

Out[5]:

	CustomerId	Surname	Geography	Gender	Age	EstimatedSalary	new_feature	Age (days)
0	15634602	Hargrave	France	Female	42	101348.88	0	15330
1	15647311	Hill	Spain	Female	41	112542.58	0	14965
2	15619304	Onio	France	Female	42	113931.57	0	15330
3	15701354	Boni	France	Female	39	93826.63	0	14235
4	15737888	Mitchell	Spain	Female	43	79084.10	0	15695

In [6]:

```
for i, row in users.iloc[:2].iterrows():
    print(row)
    print('__' * 30)
```

```
CustomerId          15634602
Surname             Hargrave
Geography             France
Gender                Female
Age                       42
EstimatedSalary    101348.88
new_feature                0
Age (days)             15330
Name: 0, dtype: object
____________________________________________________________
CustomerId          15647311
Surname                 Hill
Geography              Spain
Gender                Female
Age                       41
EstimatedSalary    112542.58
new_feature                0
Age (days)             14965
Name: 1, dtype: object
____________________________________________________________
```

In [7]:

```
age_days = []

for i, row in users.iterrows():
    age_days.append(row['Age'] * 365)

age_days[:10]
```

Out[7]:

```
[15330, 14965, 15330, 14235, 15695, 16060, 18250, 10585, 16060, 9855]
```

In [8]:

```
users['Age (days) 2'] = age_days
users.head()
```

Out[8]:

	CustomerId	Surname	Geography	Gender	Age	EstimatedSalary	new_feature	Age (days)	Age (days) 2
0	15634602	Hargrave	France	Female	42	101348.88	0	15330	15330
1	15647311	Hill	Spain	Female	41	112542.58	0	14965	14965
2	15619304	Onio	France	Female	42	113931.57	0	15330	15330
3	15701354	Boni	France	Female	39	93826.63	0	14235	14235
4	15737888	Mitchell	Spain	Female	43	79084.10	0	15695	15695

In [9]:

```python
def age_to_days(x):
    return x * 365

users['Age (days) 3'] = users['Age'].apply(age_to_days)
users.head()
```

Out[9]:

	CustomerId	Surname	Geography	Gender	Age	EstimatedSalary	new_feature	Age (days)	Age (days) 2
0	15634602	Hargrave	France	Female	42	101348.88	0	15330	15330
1	15647311	Hill	Spain	Female	41	112542.58	0	14965	14965
2	15619304	Onio	France	Female	42	113931.57	0	15330	15330
3	15701354	Boni	France	Female	39	93826.63	0	14235	14235
4	15737888	Mitchell	Spain	Female	43	79084.10	0	15695	15695

In [10]:

```python
import time
from tqdm import tqdm
tqdm.pandas()


def age_to_days(x):
    time.sleep(0.001)
    return x * 365

users['Age'].progress_apply(age_to_days)
```

```
100%|██████████| 9998/9998 [00:11<00:00, 907.22it/s]
```

Out[10]:

```
0       15330
1       14965
2       15330
3       14235
4       15695
        ...
9993    10220
9994    10585
9995    14235
9996    12775
9997    13140
Name: Age, Length: 9998, dtype: int64
```

Удаление признаков

In [11]:

```
users.drop(columns='new_feature')
users.head()
```

Out[11]:

	CustomerId	Surname	Geography	Gender	Age	EstimatedSalary	new_feature	Age (days)	Age (days) 2
0	15634602	Hargrave	France	Female	42	101348.88	0	15330	15330
1	15647311	Hill	Spain	Female	41	112542.58	0	14965	14965
2	15619304	Onio	France	Female	42	113931.57	0	15330	15330
3	15701354	Boni	France	Female	39	93826.63	0	14235	14235
4	15737888	Mitchell	Spain	Female	43	79084.10	0	15695	15695

In [12]:

```
users = users.drop(columns='new_feature')
users.head()
```

Out[12]:

	CustomerId	Surname	Geography	Gender	Age	EstimatedSalary	Age (days)	Age (days) 2	Age (days) 3
0	15634602	Hargrave	France	Female	42	101348.88	15330	15330	15330
1	15647311	Hill	Spain	Female	41	112542.58	14965	14965	14965
2	15619304	Onio	France	Female	42	113931.57	15330	15330	15330
3	15701354	Boni	France	Female	39	93826.63	14235	14235	14235
4	15737888	Mitchell	Spain	Female	43	79084.10	15695	15695	15695

In [13]:

```
users['new_feature'] = 0
```

In [14]:

```
users.drop(columns='new_feature', inplace=True)
users.head()
```

Out[14]:

	CustomerId	Surname	Geography	Gender	Age	EstimatedSalary	Age (days)	Age (days) 2	Age (days) 3
0	15634602	Hargrave	France	Female	42	101348.88	15330	15330	15330
1	15647311	Hill	Spain	Female	41	112542.58	14965	14965	14965
2	15619304	Onio	France	Female	42	113931.57	15330	15330	15330
3	15701354	Boni	France	Female	39	93826.63	14235	14235	14235
4	15737888	Mitchell	Spain	Female	43	79084.10	15695	15695	15695

In [15]:

```
users.drop(columns=['Age (days)', 'Age (days) 2', 'Age (days) 3'], inplace=True)
users.head()
```

Out[15]:

	CustomerId	Surname	Geography	Gender	Age	EstimatedSalary
0	15634602	Hargrave	France	Female	42	101348.88
1	15647311	Hill	Spain	Female	41	112542.58
2	15619304	Onio	France	Female	42	113931.57
3	15701354	Boni	France	Female	39	93826.63
4	15737888	Mitchell	Spain	Female	43	79084.10

Изменение существующих признаков

.loc

In [16]:

```
users['target'] = 0
users.head()
```

Out[16]:

	CustomerId	Surname	Geography	Gender	Age	EstimatedSalary	target
0	15634602	Hargrave	France	Female	42	101348.88	0
1	15647311	Hill	Spain	Female	41	112542.58	0
2	15619304	Onio	France	Female	42	113931.57	0
3	15701354	Boni	France	Female	39	93826.63	0
4	15737888	Mitchell	Spain	Female	43	79084.10	0

In [17]:

```
users.loc[users['Geography'] == 'France']
```

Out[17]:

	CustomerId	Surname	Geography	Gender	Age	EstimatedSalary	target
0	15634602	Hargrave	France	Female	42	101348.88	0
2	15619304	Onio	France	Female	42	113931.57	0
3	15701354	Boni	France	Female	39	93826.63	0
6	15592531	Bartlett	France	Male	50	10062.80	0
8	15792365	He	France	Male	44	74940.50	0
...	...	...	...	...	...	...	...
9993	15569266	Rahman	France	Male	28	29179.52	0
9994	15719294	Wood	France	Female	29	167773.55	0
9995	15606229	Obijiaku	France	Male	39	96270.64	0
9996	15569892	Johnstone	France	Male	35	101699.77	0
9997	15584532	Liu	France	Female	36	42085.58	0

5013 rows × 7 columns

In [18]:

```
users.loc[users['Geography'] == 'France', 'target']
```

Out[18]:

```
0       0
2       0
3       0
6       0
8       0
       ..
9993    0
9994    0
9995    0
9996    0
9997    0
Name: target, Length: 5013, dtype: int64
```

In [19]:

```
users[users['Geography'] == 'France']['target'] = 1
users.head()
```

```
<ipython-input-19-b763340dfd50>: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/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  users[users['Geography'] == 'France']['target'] = 1
```

Out[19]:

	CustomerId	Surname	Geography	Gender	Age	EstimatedSalary	target
0	15634602	Hargrave	France	Female	42	101348.88	0
1	15647311	Hill	Spain	Female	41	112542.58	0
2	15619304	Onio	France	Female	42	113931.57	0
3	15701354	Boni	France	Female	39	93826.63	0
4	15737888	Mitchell	Spain	Female	43	79084.10	0

In [20]:

```
users.loc[users['Geography'] == 'France', 'target'] = 1
users.head()
```

Out[20]:

	CustomerId	Surname	Geography	Gender	Age	EstimatedSalary	target
0	15634602	Hargrave	France	Female	42	101348.88	1
1	15647311	Hill	Spain	Female	41	112542.58	0
2	15619304	Onio	France	Female	42	113931.57	1
3	15701354	Boni	France	Female	39	93826.63	1
4	15737888	Mitchell	Spain	Female	43	79084.10	0

.replace

In [21]:

```
users['Gender'].replace({'Female': 'F', 'Male': 'M'}, inplace=True)
users.head()
```

Out[21]:

	CustomerId	Surname	Geography	Gender	Age	EstimatedSalary	target
0	15634602	Hargrave	France	F	42	101348.88	1
1	15647311	Hill	Spain	F	41	112542.58	0
2	15619304	Onio	France	F	42	113931.57	1
3	15701354	Boni	France	F	39	93826.63	1
4	15737888	Mitchell	Spain	F	43	79084.10	0

Методы агрегации

In [22]:

```
users['Age'].agg(['min', 'max'])
```

Out[22]:

```
min    18
max    92
Name: Age, dtype: int64
```

In [23]:

```
users.agg({
    'Age': ['min', 'max'],
    'EstimatedSalary': 'mean'
})
```

Out[23]:

	Age	EstimatedSalary
min	18.0	NaN
max	92.0	NaN
mean	NaN	100097.151381

In [24]:

```
users.agg(
    min_age=('Age', 'min'),
    max_age=('Age', 'max'),
    mean_salary=('EstimatedSalary', 'mean')
)
```

Out[24]:

	Age	EstimatedSalary
min_age	18.0	NaN
max_age	92.0	NaN
mean_salary	NaN	100097.151381

Методы объединения

In [25]:

```
bank = pd.read_csv('bank.csv', sep=';')
bank.head()
```

Out[25]:

	CustomerId	CreditScore	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	Exite
0	15597909	652	7	128135.99	1	1	0	
1	15687913	501	7	93244.42	1	0	1	
2	15619087	762	1	102520.37	1	1	1	
3	15596552	535	5	134542.73	1	1	1	
4	15741417	624	7	119656.45	2	1	1	

In [26]:

```
bank.shape
```

Out[26]:

```
(9895, 8)
```

In [27]:

```
merged = users.merge(bank, left_on='CustomerId', right_on='CustomerId')
merged.head()
```

Out[27]:

	CustomerId	Surname	Geography	Gender	Age	EstimatedSalary	target	CreditScore	Tenure
0	15634602	Hargrave	France	F	42	101348.88	1	619	2
1	15647311	Hill	Spain	F	41	112542.58	0	608	1
2	15619304	Onio	France	F	42	113931.57	1	502	8
3	15701354	Boni	France	F	39	93826.63	1	699	1
4	15737888	Mitchell	Spain	F	43	79084.10	0	850	2

In [28]:

```
users_id = users.set_index('CustomerId')
users_id.head()
```

Out[28]:

	Surname	Geography	Gender	Age	EstimatedSalary	target
CustomerId						
15634602	Hargrave	France	F	42	101348.88	1
15647311	Hill	Spain	F	41	112542.58	0
15619304	Onio	France	F	42	113931.57	1
15701354	Boni	France	F	39	93826.63	1
15737888	Mitchell	Spain	F	43	79084.10	0

In [29]:

```
bank_id = bank.set_index('CustomerId')
bank_id.head()
```

Out[29]:

	CreditScore	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	Exited
CustomerId							
15597909	652	7	128135.99	1	1	0	0
15687913	501	7	93244.42	1	0	1	0
15619087	762	1	102520.37	1	1	1	0
15596552	535	5	134542.73	1	1	1	1
15741417	624	7	119656.45	2	1	1	0

In [30]:

```
bank_id.join(users_id).head()
```

Out[30]:

	CreditScore	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	Exited
CustomerId							
15597909	652	7	128135.99	1	1	0	0
15687913	501	7	93244.42	1	0	1	0
15619087	762	1	102520.37	1	1	1	0
15596552	535	5	134542.73	1	1	1	1
15741417	624	7	119656.45	2	1	1	0

In [31]:

```
bank_id.join(users_id).reset_index().head()
```

Out[31]:

	CustomerId	CreditScore	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	Exite
0	15597909	652	7	128135.99	1	1	0	
1	15687913	501	7	93244.42	1	0	1	
2	15619087	762	1	102520.37	1	1	1	
3	15596552	535	5	134542.73	1	1	1	
4	15741417	624	7	119656.45	2	1	1	

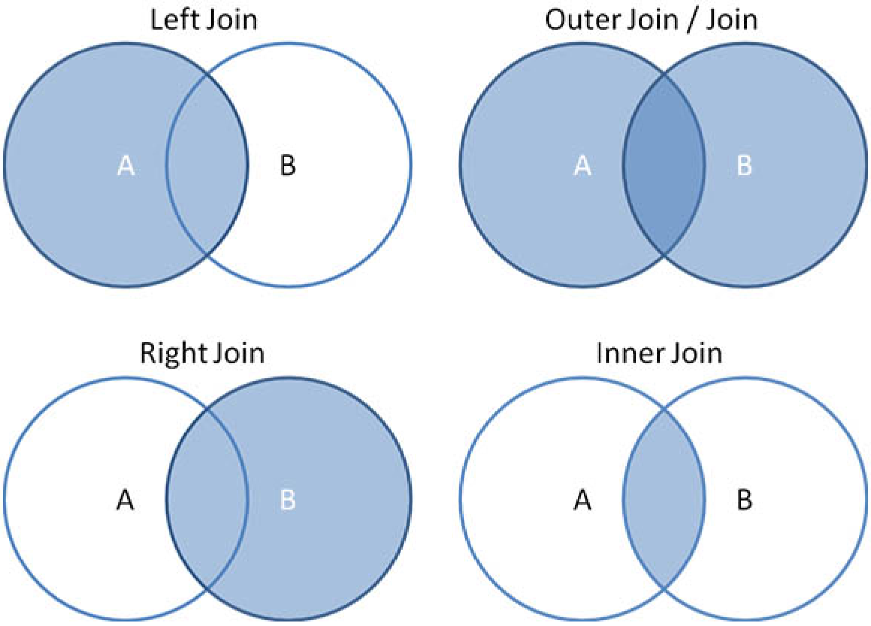
In [32]:

```
bank.shape
```

Out[32]:

```
(9895, 8)
```

Атрибут how



In [7]:
```
toy_df1 = pd.DataFrame({
    'col_1': [1, 2, 3],
    'col_2': [9, 9, 9]
})

toy_df2 = pd.DataFrame({
    'col_1': [3, 4],
    'col_3': [0, 0]
})

display(toy_df1, toy_df2)
```

	col_1	col_2
0	1	9
1	2	9
2	3	9

	col_1	col_3
0	3	0
1	4	0

In [8]:
```
toy_df1.merge(toy_df2, how='left')
```

Out[8]:

	col_1	col_2	col_3
0	1	9	NaN
1	2	9	NaN
2	3	9	0.0

In [9]:
```
toy_df1.merge(toy_df2, how='right')
```

Out[9]:

	col_1	col_2	col_3
0	3	9.0	0
1	4	NaN	0

In [10]:

```
toy_df1.merge(toy_df2, how='inner')
```

Out[10]:

	col_1	col_2	col_3
0	3	9	0

In [11]:

```
toy_df1.merge(toy_df2, how='outer')
```

Out[11]:

	col_1	col_2	col_3
0	1	9.0	NaN
1	2	9.0	NaN
2	3	9.0	0.0
3	4	NaN	0.0

left

In [33]:

```
merged_left = bank.merge(users, on='CustomerId', how='left')
merged_left.shape
```

Out[33]:

```
(9895, 14)
```

In [34]:

```
merged_left.isna().sum()
```

Out[34]:

```
CustomerId         0
CreditScore        0
Tenure             0
Balance            0
NumOfProducts      0
HasCrCard          0
IsActiveMember     0
Exited             0
Surname            2
Geography          2
Gender             2
Age                2
EstimatedSalary    2
target             2
dtype: int64
```

In [35]:

```
merged_left[merged_left['Age'].isna()]
```

Out[35]:

	CustomerId	CreditScore	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	E
6922	15682355	772	3	75075.31	2	1	0	
7360	15628319	792	4	130142.79	1	1	0	

In [36]:

```
users[users['CustomerId'] == 15682355]
```

Out[36]:

CustomerId	Surname	Geography	Gender	Age	EstimatedSalary	target

right

In [37]:
```
merged_right = bank.merge(users, on='CustomerId', how='right')
merged_right.shape
```

Out[37]:
```
(9998, 14)
```

In [38]:
```
merged_right.isna().sum()
```

Out[38]:
```
CustomerId           0
CreditScore        105
Tenure             105
Balance            105
NumOfProducts      105
HasCrCard          105
IsActiveMember     105
Exited             105
Surname              0
Geography            0
Gender               0
Age                  0
EstimatedSalary      0
target               0
dtype: int64
```

In [39]:
```
merged_right[merged_right['CreditScore'].isna()]
```

Out[39]:

	CustomerId	CreditScore	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	Ex
169	15611325	NaN	NaN	NaN	NaN	NaN	NaN	
342	15681081	NaN	NaN	NaN	NaN	NaN	NaN	
371	15774696	NaN	NaN	NaN	NaN	NaN	NaN	
609	15586585	NaN	NaN	NaN	NaN	NaN	NaN	
629	15692463	NaN	NaN	NaN	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	
9367	15785024	NaN	NaN	NaN	NaN	NaN	NaN	
9515	15792922	NaN	NaN	NaN	NaN	NaN	NaN	
9561	15810010	NaN	NaN	NaN	NaN	NaN	NaN	
9691	15754599	NaN	NaN	NaN	NaN	NaN	NaN	
9766	15795511	NaN	NaN	NaN	NaN	NaN	NaN	

105 rows × 14 columns

In [40]:
```
bank[bank['CustomerId'] == 15611325]
```

Out[40]:

CustomerId	CreditScore	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	Exited

inner

```
In [41]: merged_inner = bank.merge(users, on='CustomerId', how='inner')
         merged_inner.shape
```

```
Out[41]: (9893, 14)
```

```
In [42]: merged_inner.isna().sum()
```

```
Out[42]: CustomerId         0
         CreditScore        0
         Tenure             0
         Balance            0
         NumOfProducts      0
         HasCrCard          0
         IsActiveMember     0
         Exited             0
         Surname            0
         Geography          0
         Gender             0
         Age                0
         EstimatedSalary    0
         target             0
         dtype: int64
```

outer

```
In [43]: merged_outer = bank.merge(users, on='CustomerId', how='outer')
         merged_outer.shape
```

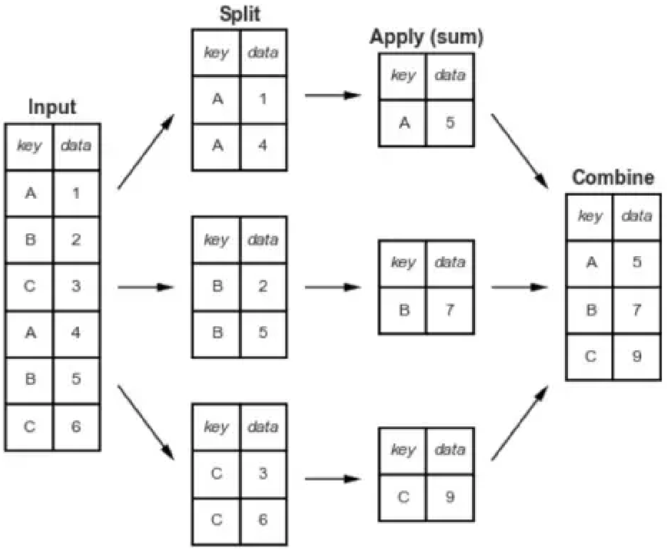
```
Out[43]: (10000, 14)
```

```
In [44]: merged_outer.isna().sum()
```

```
Out[44]: CustomerId           0
         CreditScore        105
         Tenure             105
         Balance            105
         NumOfProducts      105
         HasCrCard          105
         IsActiveMember     105
         Exited             105
         Surname              2
         Geography            2
         Gender               2
         Age                  2
         EstimatedSalary      2
         target               2
         dtype: int64
```

Методы группировок

groupby



In [45]:

```
toy_df = pd.DataFrame({
    'client_id': [1, 2, 2, 3, 1, 1],
    'item': ['chocolate', 'cheese', 'ham', 'candy', 'chair', 'book'],
    'price': [68, 280, 302, 39, 2099, 1089]
})

toy_df
```

Out[45]:

	client_id	item	price
0	1	chocolate	68
1	2	cheese	280
2	2	ham	302
3	3	candy	39
4	1	chair	2099
5	1	book	1089

In [46]:

```
grouped = toy_df.groupby('client_id')
grouped
```

Out[46]:

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x7f871b370610>
```

In [47]:

```
grouped.groups
```

Out[47]:

```
{1: [0, 4, 5], 2: [1, 2], 3: [3]}
```

In [48]:

```
grouped.sum()
```

Out[48]:

	price
client_id	
1	3256
2	582
3	39

In [49]:
```
grouped.agg({'price': ['sum', 'min', 'max']})
```

Out[49]:

	price		
	sum	min	max
client_id			
1	3256	68	2099
2	582	280	302
3	39	39	39

In [50]:
```
users.groupby('Geography').agg({'Age': ['mean'], 'EstimatedSalary': ['min']})
```

Out[50]:

	Age	EstimatedSalary
	mean	min
Geography		
France	38.513864	90.07
Germany	39.770734	11.58
Spain	38.890997	417.41

pivot_table

In [51]:
```
toy_df
```

Out[51]:

	client_id	item	price
0	1	chocolate	68
1	2	cheese	280
2	2	ham	302
3	3	candy	39
4	1	chair	2099
5	1	book	1089

In [52]:
```
toy_df.pivot_table(index='client_id',
                   values='price',
                   aggfunc='sum')
```

Out[52]:

	price
client_id	
1	3256
2	582
3	39

In [53]:
```
users.pivot_table(index='Geography',
                  aggfunc={'Age': ['mean'], 'EstimatedSalary': 'min'})
```

Out[53]:

	Age	EstimatedSalary
	mean	min
Geography		
France	38.513864	90.07
Germany	39.770734	11.58
Spain	38.890997	417.41

In [54]:

```
users.pivot_table(index='Geography',
                  columns='Gender',
                  values='EstimatedSalary',
                  aggfunc='mean',
                  margins=True,
                  margins_name='Total')
```

Out[54]:

Gender	F	M	Total
Geography			
France	99591.409159	100174.252495	99911.490489
Germany	102446.424124	99910.369711	101116.714573
Spain	100734.107475	98425.687680	99440.572281
Total	100615.282193	99665.818876	100097.151381

crosstab

In [55]:

```
pd.crosstab(index=users['Geography'],
            columns=users['Gender'])
```

Out[55]:

Gender	F	M
Geography		
France	2260	2753
Germany	1193	1315
Spain	1089	1388

In [56]:

```
pd.crosstab(index=users['Geography'],
            columns=users['Gender'],
            values=users['EstimatedSalary'],
            aggfunc='mean')
```

Out[56]:

Gender	F	M
Geography		
France	99591.409159	100174.252495
Germany	102446.424124	99910.369711
Spain	100734.107475	98425.687680

In [57]:

```
pd.crosstab(index=users['Geography'],
            columns=users['Gender'],
            normalize='all')
```

Out[57]:

Gender	F	M
Geography		
France	0.226045	0.275355
Germany	0.119324	0.131526
Spain	0.108922	0.138828

In [58]:

```
pd.crosstab(index=users['Geography'],
            columns=users['Gender'],
            normalize='index')
```

Out[58]:

Gender	F	M
Geography		
France	0.450828	0.549172
Germany	0.475678	0.524322
Spain	0.439645	0.560355

In [59]:

```
pd.crosstab(index=users['Geography'],
            columns=users['Gender'],
            normalize='columns')
```

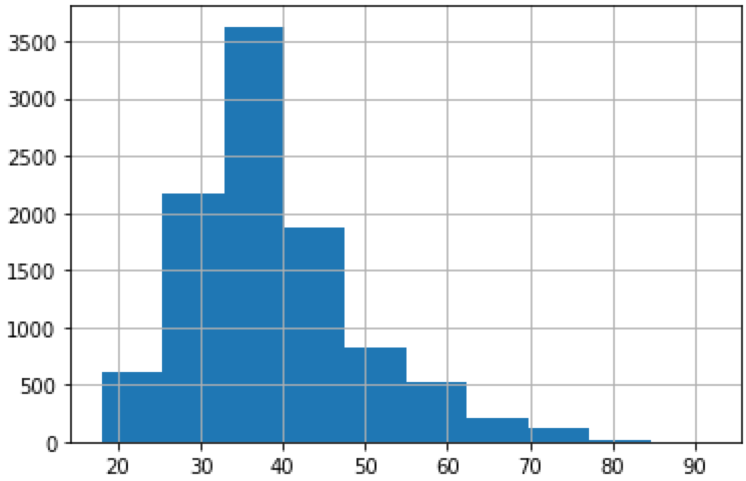
Out[59]:

Gender	F	M
Geography		
France	0.497578	0.504582
Germany	0.262660	0.241019
Spain	0.239762	0.254399

Встроенные визуализации

In [63]:

```
users['Age'].hist();
```



In [91]:

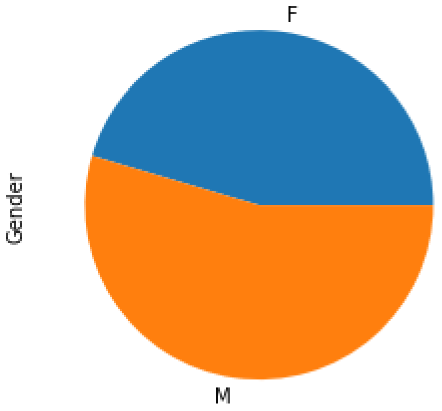
```
data = users.groupby('Gender').count()['Age']
data.name = 'Gender'
data
```

Out[91]:

```
Gender
F    4542
M    5456
Name: Gender, dtype: int64
```

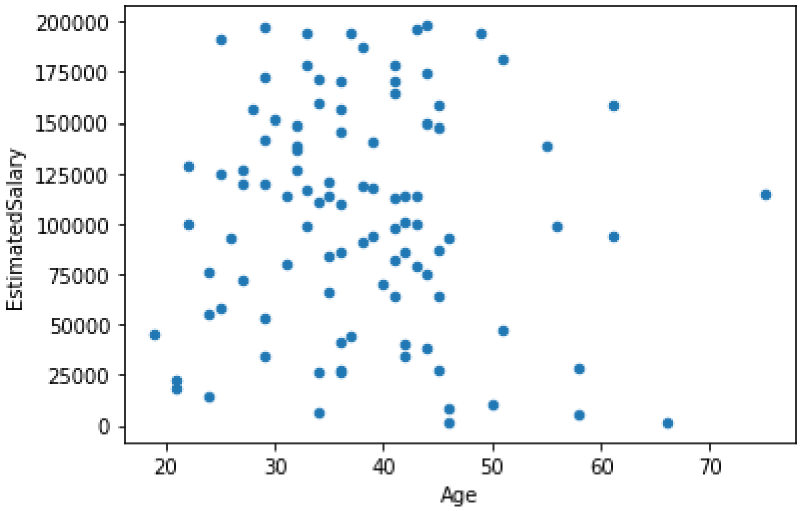
In [92]:

```
data.plot.pie(y='Gender');
```



In [93]:

```
users.iloc[:100].plot.scatter(x='Age', y='EstimatedSalary');
```



In [105]:

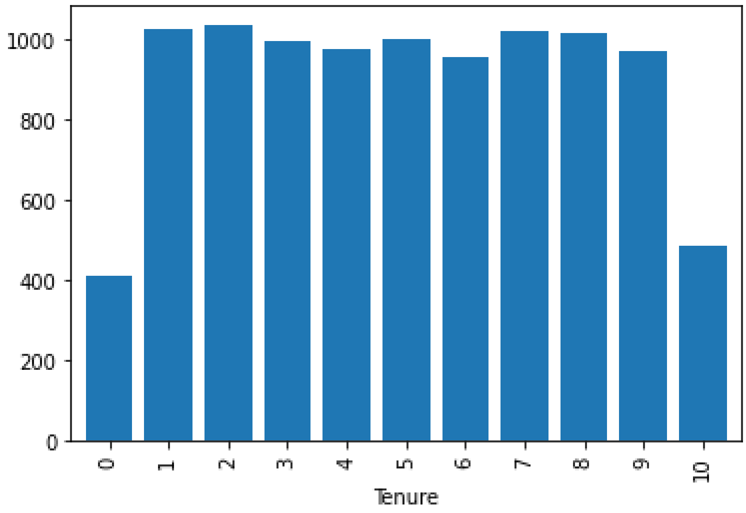
```
data = bank.groupby('Tenure').count()['Balance']
data.name = 'num_clients'
data
```

Out[105]:

```
Tenure
0      411
1     1027
2     1036
3      994
4      978
5     1000
6      957
7     1020
8     1014
9      971
10     487
Name: num_clients, dtype: int64
```

In [110]:

```
data.plot.bar(width=0.8);
```


1000
800
600
400
200
0
0
1
2
3
4
5
6
7
8
9
10
Tenure