In [52]:
```
import pymongo
from pymongo import MongoClient
client = MongoClient('mongodb://localhost:27017/')
```

In [53]:
```
#This will create a Collection/Database called department if it doesn't already exist
#This first creates a client object
coll_department = client['department']
# now to access the client object we use this syntax
department = coll_department.department
```

In [54]:
```
#Inserts department names and heads data
#This syntax is used for inserting more than one entry at one time
department.insert_many([{"dep_name": "IT", "DepartmentHead":"Jason"},
                            {"dep_name": "Admin", "DepartmentHead":"Nial"},
                            {"dep_name": "Accounts", "DepartmentHead":"Harris"}])
```

Out[54]:
```
<pymongo.results.InsertManyResult at 0x244ce2386c0>
```

In order to create the data we use will pandas dataframe. A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consists of three principal components, the data, rows, and columns.

In [55]:
```
#This will create a Collection/Database called employees if it doesn't already exist
import pandas as pd

db = client['employees']

df_employees = pd.DataFrame(list(db.employees.find()))
```

```
In [56]: df_employees.head()
```

Out[56]:

	_id	name	Department	Salary
0	60e8508a53d61dbab1c2e382	Jessica	IT	6000
1	60e8508a53d61dbab1c2e383	Joseph	IT	7000
2	60e8508a53d61dbab1c2e384	Alex	Accounts	5000
3	60e8508a53d61dbab1c2e385	Julie	IT	3000
4	60e8508a53d61dbab1c2e386	James	Admin	8000

```
In [57]: df_department = pd.DataFrame(list(department.find()))
```

```
In [58]: df_department.head()
```

Out[58]:

	_id	dep_name	DepartmentHead
0	60e867a80d2571f965a4b320	IT	Jason
1	60e867a80d2571f965a4b321	Admin	Nial
2	60e867a80d2571f965a4b322	Accounts	Harris

Left Join:

Left join uses only keys from left frame, similar to a SQL left outer join

In [59]:
```
df_employees.merge(df_department,left_on="Department",right_on="dep_name",how="left")
```

Out[59]:

	_id_x	name	Department	Salary	_id_y	dep_name	DepartmentHead
0	60e8508a53d61dbab1c2e382	Jessica	IT	6000	60e867a80d2571f965a4b320	IT	Jason
1	60e8508a53d61dbab1c2e383	Joseph	IT	7000	60e867a80d2571f965a4b320	IT	Jason
2	60e8508a53d61dbab1c2e384	Alex	Accounts	5000	60e867a80d2571f965a4b322	Accounts	Harris
3	60e8508a53d61dbab1c2e385	Julie	IT	3000	60e867a80d2571f965a4b320	IT	Jason
4	60e8508a53d61dbab1c2e386	James	Admin	8000	60e867a80d2571f965a4b321	Admin	Nial
5	60e8508a53d61dbab1c2e387	Jacob	Admin	9000	60e867a80d2571f965a4b321	Admin	Nial
6	60e8508a53d61dbab1c2e388	Kevin	IT	6500	60e867a80d2571f965a4b320	IT	Jason
7	60e8513b09c425d9fbe9f79c	Jessica	IT	6000	60e867a80d2571f965a4b320	IT	Jason
8	60e8513b09c425d9fbe9f79d	Joseph	IT	7000	60e867a80d2571f965a4b320	IT	Jason
9	60e8513b09c425d9fbe9f79e	Alex	Accounts	5000	60e867a80d2571f965a4b322	Accounts	Harris
10	60e8513b09c425d9fbe9f79f	Julie	IT	3000	60e867a80d2571f965a4b320	IT	Jason
11	60e8513b09c425d9fbe9f7a0	James	Admin	8000	60e867a80d2571f965a4b321	Admin	Nial
12	60e8513b09c425d9fbe9f7a1	Jacob	Admin	9000	60e867a80d2571f965a4b321	Admin	Nial
13	60e8513b09c425d9fbe9f7a2	Kevin	IT	6500	60e867a80d2571f965a4b320	IT	Jason

Inner Join:

Use intersection of keys from both frames, similar to a SQL inner join; preserve the order of the left keys.

In [60]:
```
df_employees.merge(df_department,left_on="Department",right_on="dep_name",how="inner")
```

Out[60]:

	_id_x	name	Department	Salary	_id_y	dep_name	DepartmentHead
0	60e8508a53d61dbab1c2e382	Jessica	IT	6000	60e867a80d2571f965a4b320	IT	Jason
1	60e8508a53d61dbab1c2e383	Joseph	IT	7000	60e867a80d2571f965a4b320	IT	Jason
2	60e8508a53d61dbab1c2e385	Julie	IT	3000	60e867a80d2571f965a4b320	IT	Jason
3	60e8508a53d61dbab1c2e388	Kevin	IT	6500	60e867a80d2571f965a4b320	IT	Jason
4	60e8513b09c425d9fbe9f79c	Jessica	IT	6000	60e867a80d2571f965a4b320	IT	Jason
5	60e8513b09c425d9fbe9f79d	Joseph	IT	7000	60e867a80d2571f965a4b320	IT	Jason
6	60e8513b09c425d9fbe9f79f	Julie	IT	3000	60e867a80d2571f965a4b320	IT	Jason
7	60e8513b09c425d9fbe9f7a2	Kevin	IT	6500	60e867a80d2571f965a4b320	IT	Jason
8	60e8508a53d61dbab1c2e384	Alex	Accounts	5000	60e867a80d2571f965a4b322	Accounts	Harris
9	60e8513b09c425d9fbe9f79e	Alex	Accounts	5000	60e867a80d2571f965a4b322	Accounts	Harris
10	60e8508a53d61dbab1c2e386	James	Admin	8000	60e867a80d2571f965a4b321	Admin	Nial
11	60e8508a53d61dbab1c2e387	Jacob	Admin	9000	60e867a80d2571f965a4b321	Admin	Nial
12	60e8513b09c425d9fbe9f7a0	James	Admin	8000	60e867a80d2571f965a4b321	Admin	Nial
13	60e8513b09c425d9fbe9f7a1	Jacob	Admin	9000	60e867a80d2571f965a4b321	Admin	Nial

To get the total number of records in the collection

In [61]:
```
pipeline = [
    {"$group": {"_id": None,"Count": {"$sum": 1}}}]
grp_employees = db.employees.aggregate(pipeline)
```

In [62]:
```
for employee in grp_employees:
  print(employee)
```

```
{'_id': None, 'Count': 14}
```

To group by department and get the total salary for each department

In [63]:
```
pipeline = [
     {"$group": {"_id": "$Department","Salary": {"$sum": "$Salary"}}}]
grp_employees = db.employees.aggregate(pipeline)
```

In [64]:
```
for employee in grp_employees:
  print(employee)
```

```
{'_id': 'Accounts', 'Salary': 10000}
{'_id': 'IT', 'Salary': 45000}
{'_id': 'Admin', 'Salary': 34000}
```

To group by department and get the average salary for each department

In [65]:
```
pipeline = [
     {"$group": {"_id": "$Department","Salary": {"$avg": "$Salary"}}}]
grp_employees = db.employees.aggregate(pipeline)
```

In [66]:
```
for employee in grp_employees:
  print(employee['_id'],"\t Average Salary ",employee['Salary'])
```

```
Accounts         Average Salary  5000.0
IT       Average Salary  5625.0
Admin    Average Salary  8500.0
```

To get the employees with the lowest salary

In [67]:
```
pipeline = [
     {"$group": {"_id": None ,"Minimum Salary": {"$min": "$Salary"}}}]
grp_employees = db.employees.aggregate(pipeline)
```

In [68]:
```
for employee in grp_employees:
    for spec_emp in db.employees.find({"Salary":employee['Minimum Salary']}):
        print("Name ",spec_emp['name'],"\nSalary ",spec_emp['Salary'])
```

```
Name  Julie
Salary  3000
Name  Julie
Salary  3000
```