#Student Database (MongoDB)  
  
#IMPORT\_REQUIRED\_MODULES  
import pymongo  
import json  
from pymongo import MongoClient  
  
client = pymongo.MongoClient("mongodb://127.0.0.1:27017")  
  
#CREATE\_DATABASE\_AND\_COLLECTION  
  
db = client["STUDENT"]  
  
DATABASE = db["DATABASE"]  
  
#OPEN\_AND\_LOAD\_FILE  
  
with open(r'C:\Users\cprat\PycharmProjects\pythonProject\students.json') as file:  
 file\_data = [json.loads(i) for i in file]  
  
#INSERT\_DATA  
  
if isinstance(file\_data, list):  
 DATABASE.insert\_many(file\_data)  
else:  
 DATABASE.insert\_one(file\_data)  
  
for i in DATABASE.find():  
 print(i)  
  
  
#1:Student name who scored maximum scores in all (exam, quiz and homework)?  
  
agg = DATABASE.aggregate([  
 {"$unwind": "$scores"},  
 {"$group":  
 {  
 "\_id": "$\_id",  
 "name": {"$first": "$name"}  
 ,  
 "Total": {"$sum": "$scores.score"},  
 }  
 },  
 {"$sort": {"Total": -1}},  
 {"$limit": 1}  
  
])   
  
  
for i in agg:  
 print(i)

ANS-{'\_id': 13, 'name': 'Jessika Dagenais', 'Total': 275.94947751518237}

#2:Students who scored below average in the exam if pass mark is 40%?  
  
agg01 = DATABASE.aggregate([  
 {"$unwind": "$scores"},  
 {"$match": {'scores.type': 'exam', "scores.score": {"$gt": 40, "$lt": 60}}  
 }  
  
])  
for i in agg01:  
 print(i)  
  
ANS----- {'\_id': 5, 'name': 'Wilburn Spiess', 'scores': {'score': 44.87186330181261, 'type': 'exam'}}

{'\_id': 10, 'name': 'Denisha Cast', 'scores': {'score': 45.61876862259409, 'type': 'exam'}}

{'\_id': 12, 'name': 'Quincy Danaher', 'scores': {'score': 54.29841278520669, 'type': 'exam'}}

{'\_id': 19, 'name': 'Gisela Levin', 'scores': {'score': 44.51211101958831, 'type': 'exam'}}

{'\_id': 20, 'name': 'Tressa Schwing', 'scores': {'score': 42.17439799514388, 'type': 'exam'}}

{'\_id': 21, 'name': 'Rosana Vales', 'scores': {'score': 46.2289476258328, 'type': 'exam'}}

{'\_id': 23, 'name': 'Tamika Schildgen', 'scores': {'score': 45.65432764125526, 'type': 'exam'}}

{'\_id': 29, 'name': 'Gwyneth Garling', 'scores': {'score': 48.36644963899371, 'type': 'exam'}}

{'\_id': 42, 'name': 'Kayce Kenyon', 'scores': {'score': 44.62441703708117, 'type': 'exam'}}

{'\_id': 45, 'name': 'Terica Brugger', 'scores': {'score': 42.1011312120801, 'type': 'exam'}}

{'\_id': 57, 'name': 'Chad Rahe', 'scores': {'score': 40.84572027366789, 'type': 'exam'}}

{'\_id': 61, 'name': 'Grady Zemke', 'scores': {'score': 51.91561300267121, 'type': 'exam'}}

{'\_id': 62, 'name': 'Vina Matsunaga', 'scores': {'score': 51.38190070034149, 'type': 'exam'}}

{'\_id': 68, 'name': 'Jenise Mcguffie', 'scores': {'score': 40.15210496060384, 'type': 'exam'}}

{'\_id': 76, 'name': 'Adrien Renda', 'scores': {'score': 57.24794864351232, 'type': 'exam'}}

{'\_id': 77, 'name': 'Efrain Claw', 'scores': {'score': 55.41266579085205, 'type': 'exam'}}

{'\_id': 89, 'name': 'Cassi Heal', 'scores': {'score': 43.04310994985133, 'type': 'exam'}}

{'\_id': 100, 'name': 'Demarcus Audette', 'scores': {'score': 47.42608580155614, 'type': 'exam'}}

{'\_id': 102, 'name': 'Mercedez Garduno', 'scores': {'score': 49.52877007656483, 'type': 'exam'}}

{'\_id': 109, 'name': 'Flora Duell', 'scores': {'score': 40.68238966626067, 'type': 'exam'}}

{'\_id': 119, 'name': 'Wilburn Spiess', 'scores': {'score': 52.36963021569788, 'type': 'exam'}}

{'\_id': 126, 'name': 'Quincy Danaher', 'scores': {'score': 40.53136904234401, 'type': 'exam'}}

{'\_id': 128, 'name': 'Alix Sherrill', 'scores': {'score': 43.67436243299881, 'type': 'exam'}}

{'\_id': 130, 'name': 'Dodie Staller', 'scores': {'score': 52.16051124848157, 'type': 'exam'}}

{'\_id': 134, 'name': 'Tressa Schwing', 'scores': {'score': 54.53947018434061, 'type': 'exam'}}

{'\_id': 142, 'name': 'Laureen Salomone', 'scores': {'score': 42.54322973844196, 'type': 'exam'}}

{'\_id': 143, 'name': 'Gwyneth Garling', 'scores': {'score': 44.29553481758053, 'type': 'exam'}}

{'\_id': 152, 'name': 'Richelle Siemers', 'scores': {'score': 52.0158789874646, 'type': 'exam'}}

{'\_id': 155, 'name': 'Aleida Elsass', 'scores': {'score': 42.89558347656537, 'type': 'exam'}}

{'\_id': 156, 'name': 'Kayce Kenyon', 'scores': {'score': 54.00824880446614, 'type': 'exam'}}

{'\_id': 164, 'name': 'Alica Pasley', 'scores': {'score': 41.3852820348269, 'type': 'exam'}}

{'\_id': 167, 'name': 'Malisa Jeanes', 'scores': {'score': 40.68676040665008, 'type': 'exam'}}

{'\_id': 169, 'name': 'Tresa Sinha', 'scores': {'score': 52.22632020277269, 'type': 'exam'}}

{'\_id': 173, 'name': 'Vinnie Auerbach', 'scores': {'score': 57.26312067710243, 'type': 'exam'}}

{'\_id': 184, 'name': 'Harriett Velarde', 'scores': {'score': 41.47988283148075, 'type': 'exam'}}

{'\_id': 185, 'name': 'Kam Senters', 'scores': {'score': 49.8822537074033, 'type': 'exam'}}

{'\_id': 193, 'name': 'Mariela Sherer', 'scores': {'score': 47.67196715489599, 'type': 'exam'}}

{'\_id': 195, 'name': 'Linnie Weigel', 'scores': {'score': 52.44578368517977, 'type': 'exam'}}

{'\_id': 196, 'name': 'Santiago Dollins', 'scores': {'score': 52.04052571137036, 'type': 'exam'}}

#4:Total and Average of the exam, quiz and homework and store them in a separate collection.  
  
TOTAL\_AVG = db.TOTAL\_AVG\_collection  
  
agg = DATABASE.aggregate([  
 {"$unwind": "$scores"},  
 {"$group":  
 {  
 "\_id": "$\_id",  
 "name": {"$first": "$name"}  
 ,  
 "Total": {"$sum": "$scores.score"},  
 "Average": {"$avg": "$scores.score"}  
 }  
 },  
 {"$sort": {"\_id": 1}}  
  
])

TTAVG = []  
  
for i in agg:  
 TTAVG.append(i)  
 print(i)

TOTAL\_AVG.insert\_many(TTAVG)

#5:New collection which consists of students who scored below average and above 40% in all the categories.  
  
AVG\_BELOW = db.belowavg\_collection  
  
agg = DATABASE.aggregate(  
[{"$match":  
 {"$expr":  
 {"$and":  
 [{"$gt": [{"$min": "$scores.score"}, 40]},  
 {"$lt": [{"$max": "$scores.score"}, 70]}  
 ]  
 }  
 }  
 }])  
  
AVGBEL = []  
for i in agg:  
 AVGBEL.append(i)  
 print(i)  
AVG\_BELOW.insert\_many(AVGBEL)

#6:New collection which consists of students who scored below the fail mark in all the categories.  
  
FAIL = db.FAILED\_collection  
agg = DATABASE.aggregate(  
 [{"$match":  
 {"$expr":  
 {"$lt": [{"$max": "$scores.score"}, 40]  
 }  
  
 }  
 }  
 ])

FAILED = []  
  
for i in agg:  
 FAILED.append(i)  
 print(i)  
  
FAIL.insert\_many(FAILED)

#7:New collection which consists of students who scored above pass mark in all the categories.  
  
PASS = db.ALLPASS\_collection  
agg = DATABASE.aggregate(  
 [{"$match":  
 {"$expr":  
 {"$gt": [{"$min": "$scores.score"}, 40]  
 }  
  
 }  
 }  
 ])  
  
PASSED = []  
  
for i in agg:  
 PASSED.append(i)  
 print(i)  
PASS.insert\_many(PASSED)