```
#first, import packages
import pymongo, certifi


#second, create client object

client = 
pymongo.MongoClient("mongodb+srv://dmjohnson:daviskate1@cluster0.gzrng.mongodb.net/
myFirstDatabase?retryWrites=true&w=majority", tlsCAFile=certifi.where())
db = client.test

col = client['sample_airbnb']['listingsAndReviews']
col2 = client['sample_supplies']['sales']
col3 = client['sample_restaurants']['restaurants']

#now you can work on the questions

#Q1
def test_mongoDB():
     for i in col.find({}).limit(1):
         return i['name']



#Q2
def get_by_bd(a,b):
   bedroom_count = 0
   for i in col.find({'bedrooms':{'$gte':a, '$lte':b}}):
       bedroom_count += 1
   return bedroom_count



#Q3
def gender_dist():
    male_count = 0
    female_count = 0
    for i in col2.find({}):
        cust = i['customer']
        gender_values = cust.values()
        gender_list = list(gender_values)
        if gender_list[0] == "M":
            male_count += 1
        else:
            female_count += 1
    return (male_count, female_count)





#Q4
def gender_stf():
    sum_male = 0
    sum_female = 0
    male_count_2 = 0
    female_count_2 = 0
    for i in col2.find({}):
        cust = i['customer']
        gender_values = cust.values()
        gender_list = list(gender_values)
        if gender_list[0] == "M":
            gender_number_male = float(gender_list[3])
            sum_male += gender_number_male
            male_count_2 += 1
        else:
            gender_number_female = float(gender_list[3])
            sum_female += gender_number_female
            female_count_2 += 1
    return (sum_male/male_count_2, sum_female/female_count_2)



#Q5
def num_prod(y,m,d):
    sum_items = 0
    for i in col2.find({}):
        items = i['items']
        if i['saleDate'].year == y and i['saleDate'].month == m and
i['saleDate'].day == d:
            for dict_item in items:
                item_values = dict_item.values()
                item_list = list(item_values)
                sum_items += item_list[3]
        else:
            continue
    return sum_items


#Q6
def num_cuisine():
    unique_cuisines = []
    for i in col3.find({}):
        if i['cuisine'] in unique_cuisines:
            continue
        else:
            unique_cuisines.append(i['cuisine'])
    return len(unique_cuisines)


#Q7
def grades(rname,rzipcode):
    list_of_grades = []
    for i in col3.find({}):
        grades = i['grades']
        zip_codes = i['address']
        zip_codes_values = zip_codes.values()
        zip_codes_list = list(zip_codes_values)
        if i['name'] == rname and zip_codes_list[3] == rzipcode:
            for dict_item in grades:
                grade_values = dict_item.values()
                grade_list = list(grade_values)
                list_of_grades.append(grade_list[1])
        else:
            continue
        return list_of_grades
```