alfa_bank_test

March 22, 2022

0.0.1 Import libriaries and update sqlite3 version

[ ]:
```
!gdown --id 1BSHIKQ7rFw5BpTq5nw1UZfjPK_7Mpnbi
!mv _sqlite3.cpython-37m-x86_64-linux-gnu.so /usr/lib/python3.7/lib-dynload/
import os
os.kill(os.getpid(), 9)
```

```
Downloading…
From: https://drive.google.com/uc?id=1BSHIKQ7rFw5BpTq5nw1UZfjPK_7Mpnbi
To: /content/_sqlite3.cpython-37m-x86_64-linux-gnu.so
100% 6.50M/6.50M [00:00<00:00, 31.3MB/s]
```

[1]:
```
import pandas as pd
import numpy as np
import sqlite3
```

0.0.2 Create tables

[2]:
```
con = sqlite3.connect('db')
```

[3]:
```
employee = pd.DataFrame(
    {
        'id_emp': [i for i in range(1,12)],
        'name_emp': ['Alex', 'Sasha', 'Gleb', 'Andrew', 'Kate', 'Jess', 'Hank',␣
 ↪'Karen', 'Rankl', 'Marsy', 'Bill'],
        'id_region': [25, 25, 36, 44, 52, 44, 63, 63, 63, 63, 44],
        'job_id': ['001', '002', '002', '004', '005', '005', '008', '008',␣
 ↪'009', '011', '001'],
        'hire_date':['2022-01-17', '2022-01-24', '2022-01-10', '2022-02-24',␣
 ↪'2021-12-04', '2022-02-11', '2022-03-15', '2022-03-15', '2022-03-15',␣
 ↪'2022-03-15', '2021-12-06']
    }
)
employee['hire_date'] = pd.to_datetime(employee['hire_date'],format='%Y-%m-%d')
employee.to_sql('employee', con, index=False, if_exists='replace')
```

```
[4]: region = pd.DataFrame(
         {
             'id_region': [25, 36, 44, 52, 34, 63],
             'name_region': ['Moscow', 'St.Petersburg', 'Kazan', 'Kamchatka',␣
      ↪'Sahalin', 'California']
         }
     )
     region.to_sql('region', con, index=False, if_exists='replace')
```

```
[5]: work = pd.DataFrame(
         {
             'job_id': ['001', '002', '003', '004', '005', '006', '007', '008',␣
      ↪'009', '011'],
             'job_name': ['analyst', 'risk-analyst', 'risk-manager', 'manager',␣
      ↪'data-engineer', 'data-analyst', 'ds', 'quant', 'ceo', 'cfo'],
             'salary': [135, 110, 150, 140, 130, 160, 200, 205, 220, 200]
         }
     )
     work.to_sql('work', con, index=False, if_exists='replace')
```

```
[7]: def select(sql):
       return pd.read_sql(sql,con)
```

0.0.3 Task 1. Download the list of employees (name_emp) who went to work in the 1st week of each month

```
[8]: sql = '''
     select
       t.*,
       (cast(strftime('%d', hire_date) as int)) as end_of_week
     from employee as t
     where end_of_week <= 7
     '''
     select(sql)
```

```
[8]:    id_emp name_emp  id_region job_id           hire_date  end_of_week
     0       5     Kate         52    005 2021-12-04 00:00:00            4
     1      11     Bill         44    001 2021-12-06 00:00:00            6
```

0.0.4 Display all regions (name_region) where the number of employees is more than 5 (2) and the sum of all salaries is more than 10 (100) thousand rubles.

[9]:

```
sql = '''
select
  r.name_region
from employee as t
left join region as r on t.id_region = r.id_region
left join work as w on t.job_id = w.job_id
  group by name_region
  having count(t.id_emp) > 2 and sum(w.salary) > 100
'''
select(sql)
```

[9]:

```
  name_region
0  California
1       Kazan
```

0.0.5 Print a list of employees (name_emp) and the ratio of their salary to the average salary of their region of residence in the same year they started working.

[13]:

```
sql = '''
with all_info as (
  select
    t.*,
    w.job_name,
    w.salary,
    r.name_region
  from employee as t
  left join region as r on t.id_region = r.id_region
  left join work as w on t.job_id = w.job_id
),

average_salary as (
  select
    (cast(strftime('%Y', t.hire_date) as int)) as year,
    t.id_region,
    avg(t.salary) as avg_salary
  from all_info as t
  group by
    year,
    t.id_region
)

select
  t.*,
  sal.avg_salary,
  t.salary / sal.avg_salary as ratio
from all_info as t
left join average_salary as sal on t.id_region = sal.id_region and sal.year =␣
 ↪(cast(strftime('%Y', t.hire_date) as int))
order by id_region
'''
select(sql)
```

[13]:

```
    id_emp name_emp  id_region job_id           hire_date       job_name  \
0        1     Alex         25    001 2022-01-17 00:00:00        analyst
1        2    Sasha         25    002 2022-01-24 00:00:00   risk-analyst
2        3     Gleb         36    002 2022-01-10 00:00:00   risk-analyst
3        4   Andrew         44    004 2022-02-24 00:00:00        manager
4        6     Jess         44    005 2022-02-11 00:00:00  data-engineer
5       11     Bill         44    001 2021-12-06 00:00:00        analyst
6        5     Kate         52    005 2021-12-04 00:00:00  data-engineer
7        7     Hank         63    008 2022-03-15 00:00:00          quant
8        8    Karen         63    008 2022-03-15 00:00:00          quant
9        9    Rankl         63    009 2022-03-15 00:00:00            ceo
10      10    Marsy         63    011 2022-03-15 00:00:00            cfo

    salary    name_region  avg_salary     ratio
0      135         Moscow       122.5  1.102041
1      110         Moscow       122.5  0.897959
2      110  St.Petersburg       110.0  1.000000
3      140          Kazan       135.0  1.037037
4      130          Kazan       135.0  0.962963
5      135          Kazan       135.0  1.000000
6      130      Kamchatka       130.0  1.000000
7      205     California       207.5  0.987952
8      205     California       207.5  0.987952
9      220     California       207.5  1.060241
10     200     California       207.5  0.963855
```

0.0.6 Display a list of employees (name_emp) and their serial number, ranking employees by salary (from high to low), in their region of residence.

[12]:

```
sql = '''
select
  t.name_emp,
  r.name_region,
  w.salary,
  row_number() over (partition by r.id_region order by w.salary desc) as sal_rnk
from employee as t
left join region as r on t.id_region = r.id_region
left join work as w on t.job_id = w.job_id
'''
select(sql)
```

```
[12]:    name_emp    name_region  salary  sal_rnk
     0       Alex         Moscow     135        1
     1      Sasha         Moscow     110        2
     2       Gleb  St.Petersburg     110        1
     3     Andrew          Kazan     140        1
     4       Bill          Kazan     135        2
     5       Jess          Kazan     130        3
     6       Kate      Kamchatka     130        1
     7      Rankl     California     220        1
     8       Hank     California     205        2
     9      Karen     California     205        3
     10     Marsy     California     200        4
```