Week 2 Practice

Dipak Bange

You will need:

- Chapter 2 (SQL Cook Book). In this notebook you will be practicing the code provided in the chapter.
- emp.csv and dept.csv from the canvas Week 1 Practice

- Step 1: Connect to database week1.db
- Step 2: Review tables
- Step 3: Delete tables

```
import sqlite3
import pandas as pd
```

STEP 1. Connect to your database named week1. You should have a file week1.db in your local directory.

```
conn = sqlite3.connect('week1.db')
c = conn.cursor()
```

STEP 2. Review tables in your database emp and dept

Every **SQLite database** has a special table named *sqlite_master*, which is a system created table

```
c.execute("select * from SQLite_master;") # if you do not need to write sql code on multiple lines, you can use single/double quotes

tables = c.fetchall()

print("Listing tables and indices from main database:")

for table in tables:
        print(table)

        print("Table Name: %s"%(table[2]))
```

```
Listing tables and indices from main database:
```

Step 3. Delete tables

DROP TABLE SQL statement drops an existing table from the SQLite database

```
c.execute('DROP TABLE emp;')
```

```
c.execute('DROP TABLE dept;')
```

Step 4. Create Tables

Option 1: insert

```
c.execute('''CREATE TABLE DEPT(
   DEPTNO INT PRIMARY KEY      NOT NULL,
   DNAME           CHAR(50) NOT NULL,
   LOC         CHAR(50)      NOT NULL
);''')
```

```
<sqlite3.Cursor at 0x7f943410c0a0>
```

```
import csv
dept_file = open("dept.csv")
dept_rows = csv.reader(dept_file)
```

```
c. executemany("INSERT INTO dept VALUES (?, ?, ?)", dept_rows)
c. execute("SELECT * FROM dept")
print(c. fetchall())
```

```
[('DEPTNO', 'DNAME', 'LOC'), (10, 'ACCOUNTING', 'NEW YORK'), (20, 'RESEARCH', 'DALLAS'), (30, 'SALES', 'CHICAGO'), (40, 'OPERATIONS', '
```

Option 2: pandas dataframe to sql

```
read_emp = pd.read_csv(r'emp.csv')
read_emp.to_sql('emp', conn, if_exists='append', index = False) # Insert the values from the csv file into the table 'emp'
```

```
c. execute("SELECT * FROM emp")
colnames = c.description
```

```
colnames_list = []
for row in colnames:
    colnames_list.append(row[0])
```

```
colnames_list
```

```
['EMPNO', 'ENAME', 'JOB', 'MGR', 'HIREDATE', 'SAL', 'COMM', 'DEPTNO']
```

```
df = pd.DataFrame(c.fetchall(), columns=colnames_list)
df.head()
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
0	7369	SMITH	CLERK	7902.0	17-Dec-05	800	NaN	20
1	7499	ALLEN	SALESMAN	7698.0	20-Feb-06	1600	300.0	30
2	7521	WARD	SALESMAN	7698.0	22-Feb-06	1250	500.0	30
3	7566	JONES	MANAGER	7839.0	2-Apr-06	2975	NaN	20
4	7654	MARTIN	SALESMAN	7698.0	28-Sep-06	1250	1400.0	30

Basics of SQL Queries

SELECT: Statement used to select rows and columns from a database.

FROM: Specifies which table in the database you want to direct your query to.

WHERE: Clause for filtering for specified value(s).

GROUP BY: Aggregating data. Needs to be used in conjunction with SQL aggregating functions like `SUM` and `COUNT`.

ORDER BY: Sorting columns in the database.

JOIN: Joins are used to combine tables with one another.

UNION, **INTERSECT/EXCEPT**: Set operations. Unioning in SQL allows one to append tables on top of one another.

Step 5. Practice Chapter 2

```
c.execute('''
select ename,job,sal
    from emp
    where deptno = 10
    order by sal asc;
''')

colnames = c.description   # gather collumn names from a new query
colnames_list = []
for row in colnames:
    colnames_list.append(row[0])

df = pd.DataFrame(c.fetchall(), columns=colnames_list)
df
```

	ENAME	JOB	SAL
0	MILLER	CLERK	1300
1	CLARK	MANAGER	2450
2	KING	PRESIDENT	5000

Exercice 1: Change the order for Salary to a descending order

```
# Exercice:
# Change the order for Salary to a descending order

c.execute('''
select ename,job,sal
    from emp
    where deptno = 10
    order by sal DESC;
''')

colnames = c.description   # gather collumn names from a new query
colnames_list = []
for row in colnames:
    colnames_list.append(row[0])

df = pd.DataFrame(c.fetchall(), columns=colnames_list)
df
```

	ENAME	JOB	SAL
0	KING	PRESIDENT	5000
1	CLARK	MANAGER	2450
2	MILLER	CLERK	1300

```
c.execute('''
select ename,job,sal
  from emp
 where deptno = 10
 order by sal desc;
 ''')
colnames = c.description   # gather collumn names from a new query
colnames_list = []
for row in colnames:
    colnames_list.append(row[0])

df = pd.DataFrame(c.fetchall(), columns=colnames_list)
df
```

	ENAME	JOB	SAL
0	KING	PRESIDENT	5000
1	CLARK	MANAGER	2450
2	MILLER	CLERK	1300

Exercice 2: Add more columns (e.g. employer number, hiring date)

```
# Exercice 2

c.execute('''
select EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO
  from emp
 where deptno = 10
 order by sal desc;
 ''')
colnames = c.description   # gather collumn names from a new query
colnames_list = []
for row in colnames:
    colnames_list.append(row[0])

df = pd.DataFrame(c.fetchall(), columns=colnames_list)
df
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
0	7839	KING	PRESIDENT	NaN	17-Nov-06	5000	None	10
1	7782	CLARK	MANAGER	7839.0	9-Jun-06	2450	None	10
2	7934	MILLER	CLERK	7782.0	23-Jan-07	1300	None	10

Exercice 3: change the WHERE condition - find the salary > 2000

```
# Exercice 3

c.execute('''
select EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO
  from emp
 where sal > 2000
 order by sal desc;
 ''')
colnames = c.description   # gather collumn names from a new query
colnames_list = []
for row in colnames:
    colnames_list.append(row[0])

df = pd.DataFrame(c.fetchall(), columns=colnames_list)
df
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
0	7839	KING	PRESIDENT	NaN	17-Nov-06	5000	None	10
1	7788	SCOTT	ANALYST	7566.0	9-Dec-07	3000	None	20
2	7902	FORD	ANALYST	7566.0	3-Dec-06	3000	None	20
3	7566	JONES	MANAGER	7839.0	2-Apr-06	2975	None	20
4	7698	BLAKE	MANAGER	7839.0	1-May-06	2850	None	30
5	7782	CLARK	MANAGER	7839.0	9-Jun-06	2450	None	10

```
c.execute('''
select ename,job
  from emp
 order by substr(job,length(job)-1);
  ''')
colnames = c.description   # gather collumn names from a new query
colnames_list = []
for row in colnames:
    colnames_list.append(row[0])

df = pd.DataFrame(c.fetchall(), columns=colnames_list)
df
```

	ENAME	JOB
0	ALLEN	SALESMAN
1	WARD	SALESMAN
2	MARTIN	SALESMAN
3	TURNER	SALESMAN
4	JONES	MANAGER
5	BLAKE	MANAGER
6	CLARK	MANAGER
7	KING	PRESIDENT
8	SMITH	CLERK
9	ADAMS	CLERK

Concatenation Example using ||

name1||' '||name2 = name1 name2 (with space between)

```
c.execute('''
select ename||' '||deptno as data
  from emp;
  ''')
colnames = c.description   # gather collumn names from a new query
colnames_list = []
for row in colnames:
    colnames_list.append(row[0])

df = pd.DataFrame(c.fetchall(), columns=colnames_list)
df
```

	data
0	SMITH 20
1	ALLEN 30
2	WARD 30
3	JONES 20
4	MARTIN 30
5	BLAKE 30
6	CLARK 10
7	SCOTT 20
8	KING 10
9	TURNER 30
10	ADAMS 20
11	JAMES 30
12	FORD 20
13	MILLER 10

Exercice 4: concatenate namae aanad job

```
# Exercice 4

c.execute('''
select ename||' '||job as data
  from emp;
  ''')
colnames = c.description   # gather collumn names from a new query
colnames_list = []
for row in colnames:
    colnames_list.append(row[0])
```

```
df = pd.DataFrame(c.fetchall(), columns=colnames_list)
df
```

	data
0	SMITH CLERK
1	ALLEN SALESMAN
2	WARD SALESMAN
3	JONES MANAGER
4	MARTIN SALESMAN
5	BLAKE MANAGER
6	CLARK MANAGER
7	SCOTT ANALYST
8	KING PRESIDENT
9	TURNER SALESMAN
10	ADAMS CLERK
11	JAMES CLERK
12	FORD ANALYST
13	MILLER CLERK

```
#TRAANSLATE is not available in sqlite
c.execute('''
select ename||' '||deptno as data
    from emp
    order by replace(data,'##########','0123456789');
  ''')
colnames = c.description   # gather collumn names from a new query
colnames_list = []
for row in colnames:
    colnames_list.append(row[0])

df = pd.DataFrame(c.fetchall(), columns=colnames_list)
df
```

	data
0	ADAMS 20
1	ALLEN 30
2	BLAKE 30
3	CLARK 10
4	FORD 20
5	JAMES 30
6	JONES 20
7	KING 10
8	MARTIN 30
9	MILLER 10
10	SCOTT 20
11	SMITH 20
12	TURNER 30
13	WARD 30

```
c.execute('''
select ename,sal,comm
    from (
  select ename,sal,comm,
         case when comm is null then 0 else 1 end as is_null
    from emp
         ) x
 order by is_null desc,comm;
```

```
  ''')
colnames = c.description   # gather collumn names from a new query
colnames_list = []
for row in colnames:
    colnames_list.append(row[0])

df = pd.DataFrame(c.fetchall(), columns=colnames_list)
df
```

	ename	sal	comm
0	TURNER	1500	0.0
1	ALLEN	1600	300.0
2	WARD	1250	500.0
3	MARTIN	1250	1400.0
4	SMITH	800	NaN
5	JONES	2975	NaN
6	BLAKE	2850	NaN
7	CLARK	2450	NaN
8	SCOTT	3000	NaN
9	KING	5000	NaN
10	ADAMS	1100	NaN
11	JAMES	950	NaN
12	FORD	3000	NaN
13	MILLER	1300	NaN

```
c.execute('''
select ename,sal,comm,
       case when comm is null then 0 else 1 end as is_null
  from emp;
  ''')
colnames = c.description   # gather collumn names from a new query
colnames_list = []
for row in colnames:
    colnames_list.append(row[0])

df = pd.DataFrame(c.fetchall(), columns=colnames_list)
df
```

	ENAME	SAL	COMM	is_null
0	SMITH	800	NaN	0
1	ALLEN	1600	300.0	1
2	WARD	1250	500.0	1
3	JONES	2975	NaN	0
4	MARTIN	1250	1400.0	1
5	BLAKE	2850	NaN	0
6	CLARK	2450	NaN	0
7	SCOTT	3000	NaN	0
8	KING	5000	NaN	0
9	TURNER	1500	0.0	1
10	ADAMS	1100	NaN	0
11	JAMES	950	NaN	0
12	FORD	3000	NaN	0
13	MILLER	1300	NaN	0

Step 6. Close the connection

```
conn.close()
```

✓ 0s completed at 12:22 PM