

SQL_Lecture_SUni

October 24, 2018

1 SQL Introduction

```
In [1]: %reload_ext autoreload
        %autoreload 2
        %load_ext sql
```

```
In [2]: %sql use suni;
```

Done.

```
Out[2]: []
```

1.1 Basic SQL

1.1.1 SFW query

```
In [3]: %%sql
        select *
        from instructor;
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[3]: [('10101', 'Srinivasan', 'Comp. Sci.', Decimal('65000.00')),
          ('12121', 'Wu', 'Finance', Decimal('90000.00')),
          ('15151', 'Mozart', 'Music', Decimal('40000.00')),
          ('22222', 'Einstein', 'Physics', Decimal('95000.00')),
          ('32343', 'El Said', 'History', Decimal('60000.00')),
          ('33456', 'Gold', 'Physics', Decimal('87000.00')),
          ('45565', 'Katz', 'Comp. Sci.', Decimal('75000.00')),
          ('58583', 'Califieri', 'History', Decimal('62000.00')),
          ('76543', 'Singh', 'Finance', Decimal('80000.00')),
          ('76766', 'Crick', 'Biology', Decimal('72000.00')),
          ('83821', 'Brandt', 'Comp. Sci.', Decimal('92000.00')),
          ('98345', 'Kim', 'Elec. Eng.', Decimal('80000.00'))]
```

1.1.2 Selection

```
In [4]: %%sql
        select *
        from instructor
        where dept_name = 'Physics';

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[4]: [('22222', 'Einstein', 'Physics', Decimal('95000.00')),
         ('33456', 'Gold', 'Physics', Decimal('87000.00'))]
```

LIKE

```
In [12]: %%sql
         select *
         from instructor
         where name like '[A-G]';

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[12]: [('22222', 'Einstein', 'Physics', Decimal('95000.00')),
          ('32343', 'El Said', 'History', Decimal('60000.00')),
          ('33456', 'Gold', 'Physics', Decimal('87000.00')),
          ('58583', 'Califieri', 'History', Decimal('62000.00')),
          ('76766', 'Crick', 'Biology', Decimal('72000.00')),
          ('83821', 'Brandt', 'Comp. Sci.', Decimal('92000.00'))]
```

```
In [18]: %%sql
         select *
         from instructor
         where name like '[^A-G]';

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[18]: [('10101', 'Srinivasan', 'Comp. Sci.', Decimal('65000.00')),
          ('12121', 'Wu', 'Finance', Decimal('90000.00')),
          ('15151', 'Mozart', 'Music', Decimal('40000.00')),
          ('45565', 'Katz', 'Comp. Sci.', Decimal('75000.00')),
          ('76543', 'Singh', 'Finance', Decimal('80000.00')),
          ('98345', 'Kim', 'Elec. Eng.', Decimal('80000.00'))]
```

```
In [22]: %%sql
         select *
         from instructor
         where name like '[A-CT-W]';
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[22]: [('12121', 'Wu', 'Finance', Decimal('90000.00')),
          ('58583', 'Califieri', 'History', Decimal('62000.00')),
          ('76766', 'Crick', 'Biology', Decimal('72000.00')),
          ('83821', 'Brandt', 'Comp. Sci.', Decimal('92000.00'))]
```

BETWEEN

```
In [15]: %%sql
         select *
         from instructor
         where salary between 75000 and 80000;
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[15]: [('45565', 'Katz', 'Comp. Sci.', Decimal('75000.00')),
          ('76543', 'Singh', 'Finance', Decimal('80000.00')),
          ('98345', 'Kim', 'Elec. Eng.', Decimal('80000.00'))]
```

IN

```
In [19]: %%sql
         select *
         from instructor
         where left(name,1) in ('C','D');
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[19]: [('58583', 'Califieri', 'History', Decimal('62000.00')),
          ('76766', 'Crick', 'Biology', Decimal('72000.00'))]
```

Multiple conditions

```
In [23]: %%sql
         select *
         from instructor
         where dept_name = 'athletics' and (salary between 50000 and 80000);
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
0 rows affected.
```

```
Out[23]: []
```

1.1.3 Projection

```
In [25]: %%sql
        select ID, name, salary
        from instructor;

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[25]: [('10101', 'Srinivasan', Decimal('65000.00')),
          ('12121', 'Wu', Decimal('90000.00')),
          ('15151', 'Mozart', Decimal('40000.00')),
          ('22222', 'Einstein', Decimal('95000.00')),
          ('32343', 'El Said', Decimal('60000.00')),
          ('33456', 'Gold', Decimal('87000.00')),
          ('45565', 'Katz', Decimal('75000.00')),
          ('58583', 'Califieri', Decimal('62000.00')),
          ('76543', 'Singh', Decimal('80000.00')),
          ('76766', 'Crick', Decimal('72000.00')),
          ('83821', 'Brandt', Decimal('92000.00')),
          ('98345', 'Kim', Decimal('80000.00'))]
```

Literals

```
In [26]: %%sql select 123456/5;

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[26]: [(24691,)]
```

```
In [27]: %%sql select 'AB';

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[27]: [('AB',)]
```

```
In [28]: %%sql
        select 'AB'
        from instructor;

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[28]: [('AB',),
          ('AB',),
          ('AB',),
          ('AB',),
          ('AB',),
          ('AB',),
          ('AB',),
          ('AB',),
          ('AB',),
          ('AB',),
          ('AB',),
          ('AB',),
          ('AB',)]
```

1.1.4 Rename

```
In [29]: %%sql
        select name as Instructor, salary [Annual Salary]
        from instructor;

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[29]: [('Srinivasan', Decimal('65000.00')),
          ('Wu', Decimal('90000.00')),
          ('Mozart', Decimal('40000.00')),
          ('Einstein', Decimal('95000.00')),
          ('El Said', Decimal('60000.00')),
          ('Gold', Decimal('87000.00')),
          ('Katz', Decimal('75000.00')),
          ('Califieri', Decimal('62000.00')),
          ('Singh', Decimal('80000.00')),
          ('Crick', Decimal('72000.00')),
          ('Brandt', Decimal('92000.00')),
          ('Kim', Decimal('80000.00'))]
```

1.1.5 Arithmetic expression

```
In [30]: %%sql
        select ID, name [Instructor Name], floor(salary/12) [Annual Salary]
        from instructor;

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[30]: [('10101', 'Srinivasan', Decimal('5416')),
          ('12121', 'Wu', Decimal('7500')),
          ('15151', 'Mozart', Decimal('3333')),
```

```
( '22222', 'Einstein', Decimal('7916')),
( '32343', 'El Said', Decimal('5000')),
( '33456', 'Gold', Decimal('7250')),
( '45565', 'Katz', Decimal('6250')),
( '58583', 'Califieri', Decimal('5166')),
( '76543', 'Singh', Decimal('6666')),
( '76766', 'Crick', Decimal('6000')),
( '83821', 'Brandt', Decimal('7666')),
( '98345', 'Kim', Decimal('6666'))]
```

1.1.6 String expression

```
In [14]: %%sql
select building + '-' + room_number [Venue]
from classroom;
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[14]: [('Packard-101',),
          ('Painter-514',),
          ('Taylor-3128',),
          ('Watson-100',),
          ('Watson-120',)]
```

1.1.7 Functions

```
In [15]: %%sql
select name, upper(left(name,1))+ '.' Initial
from instructor;
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[15]: [('Srinivasan', 'S.'),
          ('Wu', 'W.'),
          ('Mozart', 'M.'),
          ('Einstein', 'E.'),
          ('El Said', 'E.'),
          ('Gold', 'G.'),
          ('Katz', 'K.'),
          ('Califieri', 'C.'),
          ('Singh', 'S.'),
          ('Crick', 'C.'),
          ('Brandt', 'B.'),
          ('Kim', 'K.')]

```

DISTINCT

```
In [31]: %%sql
        select dept_name
        from instructor;

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[31]: [('Comp. Sci.',),
          ('Finance',),
          ('Music',),
          ('Physics',),
          ('History',),
          ('Physics',),
          ('Comp. Sci.',),
          ('History',),
          ('Finance',),
          ('Biology',),
          ('Comp. Sci.',),
          ('Elec. Eng.',)]
```

```
In [32]: %%sql
        select distinct dept_name
        from instructor;

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[32]: [('Biology',),
          ('Comp. Sci.',),
          ('Elec. Eng.',),
          ('Finance',),
          ('History',),
          ('Music',),
          ('Physics',)]
```

1.1.8 Ordering results

order by

```
In [35]: %%sql
        select name, dept_name, salary
        from instructor
        where salary >= 50000 and salary <= 80000
        order by dept_name, salary desc;
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out [35]: [('Califieri', 'History', Decimal('62000.00')),
          ('El Said', 'History', Decimal('60000.00')),
          ('Singh', 'Finance', Decimal('80000.00')),
          ('Kim', 'Elec. Eng.', Decimal('80000.00')),
          ('Katz', 'Comp. Sci.', Decimal('75000.00')),
          ('Srinivasan', 'Comp. Sci.', Decimal('65000.00')),
          ('Crick', 'Biology', Decimal('72000.00'))]
```

order by using column positions

```
In [34]: %%sql
        select name, dept_name, salary
        from instructor
        where salary >= 50000 and salary <= 80000
        order by 2, 3 desc;
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out [34]: [('Crick', 'Biology', Decimal('72000.00')),
          ('Katz', 'Comp. Sci.', Decimal('75000.00')),
          ('Srinivasan', 'Comp. Sci.', Decimal('65000.00')),
          ('Kim', 'Elec. Eng.', Decimal('80000.00')),
          ('Singh', 'Finance', Decimal('80000.00')),
          ('Califieri', 'History', Decimal('62000.00')),
          ('El Said', 'History', Decimal('60000.00'))]
```

getting top 5 rows

```
In [38]: %%sql
        select top 5 name, dept_name, salary
        from instructor
        order by salary desc;
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out [38]: [('Einstein', 'Physics', Decimal('95000.00')),
          ('Brandt', 'Comp. Sci.', Decimal('92000.00')),
          ('Wu', 'Finance', Decimal('90000.00')),
          ('Gold', 'Physics', Decimal('87000.00')),
          ('Singh', 'Finance', Decimal('80000.00'))]
```


getting top 5 percent of rows

```
In [39]: %%sql
        select top 5 percent name, dept_name, salary
        from instructor
        order by salary desc;

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[39]: [('Einstein', 'Physics', Decimal('95000.00'))]
```

with ties

```
In [40]: %%sql
        select top 5 percent with ties name, dept_name, salary
        from instructor
        order by salary desc;

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[40]: [('Einstein', 'Physics', Decimal('95000.00'))]
```

fetch first five rows

```
In [41]: %%sql
        select name, dept_name, salary
        from instructor
        where salary >= 50000 and salary <= 80000
        order by salary desc
        offset 0 rows
        fetch first 5 rows only;

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[41]: [('Singh', 'Finance', Decimal('80000.00')),
          ('Kim', 'Elec. Eng.', Decimal('80000.00')),
          ('Katz', 'Comp. Sci.', Decimal('75000.00')),
          ('Crick', 'Biology', Decimal('72000.00')),
          ('Srinivasan', 'Comp. Sci.', Decimal('65000.00'))]
```

fetch 6th to 10th rows

```
In [42]: %%sql
        select name, dept_name, salary
        from instructor
        where salary >= 50000 and salary <= 80000
        order by salary desc
            offset 5 rows
            fetch next 5 rows only;

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[42]: [('Califieri', 'History', Decimal('62000.00')),
          ('El Said', 'History', Decimal('60000.00'))]
```

1.2 Join Queries

1.2.1 Cartesian product

```
In [44]: %%sql
        select i.ID, i.name, i.salary, d.ID, d.name, d.salary
        from instructor i, instructor d;

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[44]: [('10101', 'Srinivasan', Decimal('65000.00'), '10101', 'Srinivasan', Decimal('65000.00')),
          ('12121', 'Wu', Decimal('90000.00'), '10101', 'Srinivasan', Decimal('65000.00')),
          ('15151', 'Mozart', Decimal('40000.00'), '10101', 'Srinivasan', Decimal('65000.00')),
          ('22222', 'Einstein', Decimal('95000.00'), '10101', 'Srinivasan', Decimal('65000.00')),
          ('32343', 'El Said', Decimal('60000.00'), '10101', 'Srinivasan', Decimal('65000.00')),
          ('33456', 'Gold', Decimal('87000.00'), '10101', 'Srinivasan', Decimal('65000.00')),
          ('45565', 'Katz', Decimal('75000.00'), '10101', 'Srinivasan', Decimal('65000.00')),
          ('58583', 'Califieri', Decimal('62000.00'), '10101', 'Srinivasan', Decimal('65000.00')),
          ('76543', 'Singh', Decimal('80000.00'), '10101', 'Srinivasan', Decimal('65000.00')),
          ('76766', 'Crick', Decimal('72000.00'), '10101', 'Srinivasan', Decimal('65000.00')),
          ('83821', 'Brandt', Decimal('92000.00'), '10101', 'Srinivasan', Decimal('65000.00')),
          ('98345', 'Kim', Decimal('80000.00'), '10101', 'Srinivasan', Decimal('65000.00')),
          ('10101', 'Srinivasan', Decimal('65000.00'), '12121', 'Wu', Decimal('90000.00')),
          ('12121', 'Wu', Decimal('90000.00'), '12121', 'Wu', Decimal('90000.00')),
          ('15151', 'Mozart', Decimal('40000.00'), '12121', 'Wu', Decimal('90000.00')),
          ('22222', 'Einstein', Decimal('95000.00'), '12121', 'Wu', Decimal('90000.00')),
          ('32343', 'El Said', Decimal('60000.00'), '12121', 'Wu', Decimal('90000.00')),
          ('33456', 'Gold', Decimal('87000.00'), '12121', 'Wu', Decimal('90000.00')),
          ('45565', 'Katz', Decimal('75000.00'), '12121', 'Wu', Decimal('90000.00')),
          ('58583', 'Califieri', Decimal('62000.00'), '12121', 'Wu', Decimal('90000.00'))]
```

('76543', 'Singh', Decimal('80000.00'), '12121', 'Wu', Decimal('90000.00')),
 ('76766', 'Crick', Decimal('72000.00'), '12121', 'Wu', Decimal('90000.00')),
 ('83821', 'Brandt', Decimal('92000.00'), '12121', 'Wu', Decimal('90000.00')),
 ('98345', 'Kim', Decimal('80000.00'), '12121', 'Wu', Decimal('90000.00')),
 ('10101', 'Srinivasan', Decimal('65000.00'), '15151', 'Mozart', Decimal('40000.00')),
 ('12121', 'Wu', Decimal('90000.00'), '15151', 'Mozart', Decimal('40000.00')),
 ('15151', 'Mozart', Decimal('40000.00'), '15151', 'Mozart', Decimal('40000.00')),
 ('22222', 'Einstein', Decimal('95000.00'), '15151', 'Mozart', Decimal('40000.00')),
 ('32343', 'El Said', Decimal('60000.00'), '15151', 'Mozart', Decimal('40000.00')),
 ('33456', 'Gold', Decimal('87000.00'), '15151', 'Mozart', Decimal('40000.00')),
 ('45565', 'Katz', Decimal('75000.00'), '15151', 'Mozart', Decimal('40000.00')),
 ('58583', 'Califieri', Decimal('62000.00'), '15151', 'Mozart', Decimal('40000.00')),
 ('76543', 'Singh', Decimal('80000.00'), '15151', 'Mozart', Decimal('40000.00')),
 ('76766', 'Crick', Decimal('72000.00'), '15151', 'Mozart', Decimal('40000.00')),
 ('83821', 'Brandt', Decimal('92000.00'), '15151', 'Mozart', Decimal('40000.00')),
 ('98345', 'Kim', Decimal('80000.00'), '15151', 'Mozart', Decimal('40000.00')),
 ('10101', 'Srinivasan', Decimal('65000.00'), '22222', 'Einstein', Decimal('95000.00')),
 ('12121', 'Wu', Decimal('90000.00'), '22222', 'Einstein', Decimal('95000.00')),
 ('15151', 'Mozart', Decimal('40000.00'), '22222', 'Einstein', Decimal('95000.00')),
 ('22222', 'Einstein', Decimal('95000.00'), '22222', 'Einstein', Decimal('95000.00')),
 ('32343', 'El Said', Decimal('60000.00'), '22222', 'Einstein', Decimal('95000.00')),
 ('33456', 'Gold', Decimal('87000.00'), '22222', 'Einstein', Decimal('95000.00')),
 ('45565', 'Katz', Decimal('75000.00'), '22222', 'Einstein', Decimal('95000.00')),
 ('58583', 'Califieri', Decimal('62000.00'), '22222', 'Einstein', Decimal('95000.00')),
 ('76543', 'Singh', Decimal('80000.00'), '22222', 'Einstein', Decimal('95000.00')),
 ('76766', 'Crick', Decimal('72000.00'), '22222', 'Einstein', Decimal('95000.00')),
 ('83821', 'Brandt', Decimal('92000.00'), '22222', 'Einstein', Decimal('95000.00')),
 ('98345', 'Kim', Decimal('80000.00'), '22222', 'Einstein', Decimal('95000.00')),
 ('10101', 'Srinivasan', Decimal('65000.00'), '32343', 'El Said', Decimal('60000.00')),
 ('12121', 'Wu', Decimal('90000.00'), '32343', 'El Said', Decimal('60000.00')),
 ('15151', 'Mozart', Decimal('40000.00'), '32343', 'El Said', Decimal('60000.00')),
 ('22222', 'Einstein', Decimal('95000.00'), '32343', 'El Said', Decimal('60000.00')),
 ('32343', 'El Said', Decimal('60000.00'), '32343', 'El Said', Decimal('60000.00')),
 ('33456', 'Gold', Decimal('87000.00'), '32343', 'El Said', Decimal('60000.00')),
 ('45565', 'Katz', Decimal('75000.00'), '32343', 'El Said', Decimal('60000.00')),
 ('58583', 'Califieri', Decimal('62000.00'), '32343', 'El Said', Decimal('60000.00')),
 ('76543', 'Singh', Decimal('80000.00'), '32343', 'El Said', Decimal('60000.00')),
 ('76766', 'Crick', Decimal('72000.00'), '32343', 'El Said', Decimal('60000.00')),
 ('83821', 'Brandt', Decimal('92000.00'), '32343', 'El Said', Decimal('60000.00')),
 ('98345', 'Kim', Decimal('80000.00'), '32343', 'El Said', Decimal('60000.00')),
 ('10101', 'Srinivasan', Decimal('65000.00'), '33456', 'Gold', Decimal('87000.00')),
 ('12121', 'Wu', Decimal('90000.00'), '33456', 'Gold', Decimal('87000.00')),
 ('15151', 'Mozart', Decimal('40000.00'), '33456', 'Gold', Decimal('87000.00')),
 ('22222', 'Einstein', Decimal('95000.00'), '33456', 'Gold', Decimal('87000.00')),
 ('32343', 'El Said', Decimal('60000.00'), '33456', 'Gold', Decimal('87000.00')),
 ('33456', 'Gold', Decimal('87000.00'), '33456', 'Gold', Decimal('87000.00')),
 ('45565', 'Katz', Decimal('75000.00'), '33456', 'Gold', Decimal('87000.00')),
 ('58583', 'Califieri', Decimal('62000.00'), '33456', 'Gold', Decimal('87000.00')),

('76543', 'Singh', Decimal('80000.00'), '33456', 'Gold', Decimal('87000.00')),
('76766', 'Crick', Decimal('72000.00'), '33456', 'Gold', Decimal('87000.00')),
('83821', 'Brandt', Decimal('92000.00'), '33456', 'Gold', Decimal('87000.00')),
('98345', 'Kim', Decimal('80000.00'), '33456', 'Gold', Decimal('87000.00')),
('10101', 'Srinivasan', Decimal('65000.00'), '45565', 'Katz', Decimal('75000.00')),
('12121', 'Wu', Decimal('90000.00'), '45565', 'Katz', Decimal('75000.00')),
('15151', 'Mozart', Decimal('40000.00'), '45565', 'Katz', Decimal('75000.00')),
('22222', 'Einstein', Decimal('95000.00'), '45565', 'Katz', Decimal('75000.00')),
('32343', 'El Said', Decimal('60000.00'), '45565', 'Katz', Decimal('75000.00')),
('33456', 'Gold', Decimal('87000.00'), '45565', 'Katz', Decimal('75000.00')),
('45565', 'Katz', Decimal('75000.00'), '45565', 'Katz', Decimal('75000.00')),
('58583', 'Califieri', Decimal('62000.00'), '45565', 'Katz', Decimal('75000.00')),
('76543', 'Singh', Decimal('80000.00'), '45565', 'Katz', Decimal('75000.00')),
('76766', 'Crick', Decimal('72000.00'), '45565', 'Katz', Decimal('75000.00')),
('83821', 'Brandt', Decimal('92000.00'), '45565', 'Katz', Decimal('75000.00')),
('98345', 'Kim', Decimal('80000.00'), '45565', 'Katz', Decimal('75000.00')),
('10101', 'Srinivasan', Decimal('65000.00'), '58583', 'Califieri', Decimal('62000.00')),
('12121', 'Wu', Decimal('90000.00'), '58583', 'Califieri', Decimal('62000.00')),
('15151', 'Mozart', Decimal('40000.00'), '58583', 'Califieri', Decimal('62000.00')),
('22222', 'Einstein', Decimal('95000.00'), '58583', 'Califieri', Decimal('62000.00')),
('32343', 'El Said', Decimal('60000.00'), '58583', 'Califieri', Decimal('62000.00')),
('33456', 'Gold', Decimal('87000.00'), '58583', 'Califieri', Decimal('62000.00')),
('45565', 'Katz', Decimal('75000.00'), '58583', 'Califieri', Decimal('62000.00')),
('58583', 'Califieri', Decimal('62000.00'), '58583', 'Califieri', Decimal('62000.00')),
('76543', 'Singh', Decimal('80000.00'), '58583', 'Califieri', Decimal('62000.00')),
('76766', 'Crick', Decimal('72000.00'), '58583', 'Califieri', Decimal('62000.00')),
('83821', 'Brandt', Decimal('92000.00'), '58583', 'Califieri', Decimal('62000.00')),
('98345', 'Kim', Decimal('80000.00'), '58583', 'Califieri', Decimal('62000.00')),
('10101', 'Srinivasan', Decimal('65000.00'), '76543', 'Singh', Decimal('80000.00')),
('12121', 'Wu', Decimal('90000.00'), '76543', 'Singh', Decimal('80000.00')),
('15151', 'Mozart', Decimal('40000.00'), '76543', 'Singh', Decimal('80000.00')),
('22222', 'Einstein', Decimal('95000.00'), '76543', 'Singh', Decimal('80000.00')),
('32343', 'El Said', Decimal('60000.00'), '76543', 'Singh', Decimal('80000.00')),
('33456', 'Gold', Decimal('87000.00'), '76543', 'Singh', Decimal('80000.00')),
('45565', 'Katz', Decimal('75000.00'), '76543', 'Singh', Decimal('80000.00')),
('58583', 'Califieri', Decimal('62000.00'), '76543', 'Singh', Decimal('80000.00')),
('76543', 'Singh', Decimal('80000.00'), '76543', 'Singh', Decimal('80000.00')),
('76766', 'Crick', Decimal('72000.00'), '76543', 'Singh', Decimal('80000.00')),
('83821', 'Brandt', Decimal('92000.00'), '76543', 'Singh', Decimal('80000.00')),
('98345', 'Kim', Decimal('80000.00'), '76543', 'Singh', Decimal('80000.00')),
('10101', 'Srinivasan', Decimal('65000.00'), '76766', 'Crick', Decimal('72000.00')),
('12121', 'Wu', Decimal('90000.00'), '76766', 'Crick', Decimal('72000.00')),
('15151', 'Mozart', Decimal('40000.00'), '76766', 'Crick', Decimal('72000.00')),
('22222', 'Einstein', Decimal('95000.00'), '76766', 'Crick', Decimal('72000.00')),
('32343', 'El Said', Decimal('60000.00'), '76766', 'Crick', Decimal('72000.00')),
('33456', 'Gold', Decimal('87000.00'), '76766', 'Crick', Decimal('72000.00')),
('45565', 'Katz', Decimal('75000.00'), '76766', 'Crick', Decimal('72000.00')),
('58583', 'Califieri', Decimal('62000.00'), '76766', 'Crick', Decimal('72000.00')),

```
( '76543', 'Singh', Decimal('80000.00'), '76766', 'Crick', Decimal('72000.00')),
( '76766', 'Crick', Decimal('72000.00'), '76766', 'Crick', Decimal('72000.00')),
( '83821', 'Brandt', Decimal('92000.00'), '76766', 'Crick', Decimal('72000.00')),
( '98345', 'Kim', Decimal('80000.00'), '76766', 'Crick', Decimal('72000.00')),
( '10101', 'Srinivasan', Decimal('65000.00'), '83821', 'Brandt', Decimal('92000.00')),
( '12121', 'Wu', Decimal('90000.00'), '83821', 'Brandt', Decimal('92000.00')),
( '15151', 'Mozart', Decimal('40000.00'), '83821', 'Brandt', Decimal('92000.00')),
( '22222', 'Einstein', Decimal('95000.00'), '83821', 'Brandt', Decimal('92000.00')),
( '32343', 'El Said', Decimal('60000.00'), '83821', 'Brandt', Decimal('92000.00')),
( '33456', 'Gold', Decimal('87000.00'), '83821', 'Brandt', Decimal('92000.00')),
( '45565', 'Katz', Decimal('75000.00'), '83821', 'Brandt', Decimal('92000.00')),
( '58583', 'Califieri', Decimal('62000.00'), '83821', 'Brandt', Decimal('92000.00')),
( '76543', 'Singh', Decimal('80000.00'), '83821', 'Brandt', Decimal('92000.00')),
( '76766', 'Crick', Decimal('72000.00'), '83821', 'Brandt', Decimal('92000.00')),
( '83821', 'Brandt', Decimal('92000.00'), '83821', 'Brandt', Decimal('92000.00')),
( '98345', 'Kim', Decimal('80000.00'), '83821', 'Brandt', Decimal('92000.00')),
( '10101', 'Srinivasan', Decimal('65000.00'), '98345', 'Kim', Decimal('80000.00')),
( '12121', 'Wu', Decimal('90000.00'), '98345', 'Kim', Decimal('80000.00')),
( '15151', 'Mozart', Decimal('40000.00'), '98345', 'Kim', Decimal('80000.00')),
( '22222', 'Einstein', Decimal('95000.00'), '98345', 'Kim', Decimal('80000.00')),
( '32343', 'El Said', Decimal('60000.00'), '98345', 'Kim', Decimal('80000.00')),
( '33456', 'Gold', Decimal('87000.00'), '98345', 'Kim', Decimal('80000.00')),
( '45565', 'Katz', Decimal('75000.00'), '98345', 'Kim', Decimal('80000.00')),
( '58583', 'Califieri', Decimal('62000.00'), '98345', 'Kim', Decimal('80000.00')),
( '76543', 'Singh', Decimal('80000.00'), '98345', 'Kim', Decimal('80000.00')),
( '76766', 'Crick', Decimal('72000.00'), '98345', 'Kim', Decimal('80000.00')),
( '83821', 'Brandt', Decimal('92000.00'), '98345', 'Kim', Decimal('80000.00')),
( '98345', 'Kim', Decimal('80000.00'), '98345', 'Kim', Decimal('80000.00'))]
```

1.2.2 Theta join

In [26]: `%%sql`

```
select distinct i.salary
from instructor i join instructor d on
    i.salary < d.salary;
```

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.

Out[26]: [(Decimal('40000.00'),),
(Decimal('60000.00'),),
(Decimal('62000.00'),),
(Decimal('65000.00'),),
(Decimal('72000.00'),),
(Decimal('75000.00'),),
(Decimal('80000.00'),),
(Decimal('87000.00'),),

```
(Decimal('90000.00'),),
(Decimal('92000.00'),)]
```

```
In [45]: %%sql
select distinct i.salary
from instructor i, instructor d
where i.salary < d.salary;
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[45]: [(Decimal('40000.00'),),
(Decimal('60000.00'),),
(Decimal('62000.00'),),
(Decimal('65000.00'),),
(Decimal('72000.00'),),
(Decimal('75000.00'),),
(Decimal('80000.00'),),
(Decimal('87000.00'),),
(Decimal('90000.00'),),
(Decimal('92000.00'),)]
```

1.2.3 Equi-join

```
In [28]: %%sql
select i.ID, i.name, i.dept_name, t.course_id
from instructor i inner join teaches t on
      i.ID = t.ID
where i.dept_name = 'Comp. Sci.';
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[28]: [('10101', 'Srinivasan', 'Comp. Sci.', 'CS-101'),
('10101', 'Srinivasan', 'Comp. Sci.', 'CS-315'),
('10101', 'Srinivasan', 'Comp. Sci.', 'CS-347'),
('45565', 'Katz', 'Comp. Sci.', 'CS-101'),
('45565', 'Katz', 'Comp. Sci.', 'CS-319'),
('83821', 'Brandt', 'Comp. Sci.', 'CS-190'),
('83821', 'Brandt', 'Comp. Sci.', 'CS-190'),
('83821', 'Brandt', 'Comp. Sci.', 'CS-319')]
```

```
In [29]: %%sql
select i.ID, i.name, i.dept_name, t.course_id
from instructor i, teaches t
where i.ID = t.ID and i.dept_name = 'Comp. Sci.';
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out [29]: [('10101', 'Srinivasan', 'Comp. Sci.', 'CS-101'),
          ('10101', 'Srinivasan', 'Comp. Sci.', 'CS-315'),
          ('10101', 'Srinivasan', 'Comp. Sci.', 'CS-347'),
          ('45565', 'Katz', 'Comp. Sci.', 'CS-101'),
          ('45565', 'Katz', 'Comp. Sci.', 'CS-319'),
          ('83821', 'Brandt', 'Comp. Sci.', 'CS-190'),
          ('83821', 'Brandt', 'Comp. Sci.', 'CS-190'),
          ('83821', 'Brandt', 'Comp. Sci.', 'CS-319')]
```

1.2.4 Outer Joins

```
In [46]: %%sql
        select i.ID, i.name, i.dept_name, t.course_id
        from instructor i left outer join teaches t
        on i.ID = t.ID;
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out [46]: [('10101', 'Srinivasan', 'Comp. Sci.', 'CS-101'),
          ('10101', 'Srinivasan', 'Comp. Sci.', 'CS-315'),
          ('10101', 'Srinivasan', 'Comp. Sci.', 'CS-347'),
          ('12121', 'Wu', 'Finance', 'FIN-201'),
          ('15151', 'Mozart', 'Music', 'MU-199'),
          ('22222', 'Einstein', 'Physics', 'PHY-101'),
          ('32343', 'El Said', 'History', 'HIS-351'),
          ('33456', 'Gold', 'Physics', None),
          ('45565', 'Katz', 'Comp. Sci.', 'CS-101'),
          ('45565', 'Katz', 'Comp. Sci.', 'CS-319'),
          ('58583', 'Califieri', 'History', None),
          ('76543', 'Singh', 'Finance', None),
          ('76766', 'Crick', 'Biology', 'BIO-101'),
          ('76766', 'Crick', 'Biology', 'BIO-301'),
          ('83821', 'Brandt', 'Comp. Sci.', 'CS-190'),
          ('83821', 'Brandt', 'Comp. Sci.', 'CS-190'),
          ('83821', 'Brandt', 'Comp. Sci.', 'CS-319'),
          ('98345', 'Kim', 'Elec. Eng.', 'EE-181')]
```

1.3 Set Operations

1.3.1 Intersect

Courses taught in both Fall 2009 and Spring 2010

```
In [47]: %%sql
        (select distinct course_id
         from section s
         where s.semester = 'Fall' and s.year = 2009)
        intersect
        (select distinct course_id
         from section s
         where s.semester = 'Spring' and s.year = 2010);

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[47]: [('CS-101',)]
```

1.3.2 Except

Courses taught in Fall 2009 but not in Spring 2010

```
In [50]: %%sql
        (select distinct course_id
         from section s
         where s.semester = 'Fall' and s.year = 2009)
        except
        (select distinct course_id
         from section s
         where s.semester = 'Spring' and s.year = 2010);

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[50]: [('CS-347',), ('PHY-101',)]
```

Find the highest salary of all instructors

```
In [51]: %%sql
        select distinct salary
        from instructor
        except
        select distinct i.salary
        from instructor i join instructor i2 on
            i.salary < i2.salary;

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[51]: [(Decimal('95000.00'),)]
```


1.3.3 Union

```
In [48]: %%sql
        select 5 from instructor
        union
        select 5;
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[48]: [(5,)]
```

```
In [5]: %%sql
        (select course_id
         from section
         where semester = 'Fall' and year= 2009)
        union
        (select course_id
         from section
         where semester = 'Spring' and year= 2010);
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[5]: [('CS-101',),
         ('CS-315',),
         ('CS-319',),
         ('CS-347',),
         ('FIN-201',),
         ('HIS-351',),
         ('MU-199',),
         ('PHY-101',)]
```

1.3.4 Union All

```
In [49]: %%sql
        select 5 from instructor
        union all
        select 5;
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[49]: [(5,), (5,), (5,), (5,), (5,), (5,), (5,), (5,), (5,), (5,), (5,), (5,), (5,)]
```

```
In [6]: %%sql
        (select course_id
```

```

from section
where semester = 'Fall' and year= 2009)
union all
(select course_id
from section
where semester = 'Spring' and year= 2010);

```

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.

```

Out[6]: [('CS-101',),
         ('CS-347',),
         ('PHY-101',),
         ('CS-101',),
         ('CS-315',),
         ('CS-319',),
         ('CS-319',),
         ('FIN-201',),
         ('HIS-351',),
         ('MU-199',)]

```

1.4 Nested Queries

1.4.1 Subqueries in WHERE clause

Test set membership Find course taught in Fall 2009 and Spring 2010

```

In [58]: %%sql
select distinct course_id
from section
where semester = 'Fall' and year = 2009;

```

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.

```

Out[58]: [('CS-101',), ('CS-347',), ('PHY-101',)]

```

```

In [54]: %%sql
select course_id
from section
where semester = 'Spring' and year = 2010

```

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.

```

Out[54]: [('CS-101',),
         ('CS-315',),

```

```
( 'CS-319', ),
( 'CS-319', ),
( 'FIN-201', ),
( 'HIS-351', ),
( 'MU-199', )]
```

```
In [55]: %%sql
select distinct course_id
from section
where semester = 'Fall' and year = 2009
      and course_id in (
          select course_id
          from section
          where semester = 'Spring' and year = 2010
      );
```

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.

```
Out [55]: [( 'CS-101', )]
```

Find course taught in Fall 2009 but not in Spring 2010

```
In [56]: %%sql
select distinct course_id
from section
where semester = 'Fall' and year = 2009
      and course_id not in (
          select course_id
          from section
          where semester = 'Spring' and year = 2010
      );
```

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.

```
Out [56]: [( 'CS-347', ), ( 'PHY-101', )]
```

Set Comparison Find names of instructors with salary greater than that of some (at least one) instructor in the Biology department.

```
In [38]: %%sql
select name
from instructor
where salary > some(
    select salary from instructor
    where dept_name = 'Biology');
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[38]: [('Wu',),
          ('Einstein',),
          ('Gold',),
          ('Katz',),
          ('Singh',),
          ('Brandt',),
          ('Kim',)]
```

Test for empty relations Find courses taught in both Fall 2009 and Spring 2010

```
In [39]: %%sql
select course_id
from section s
where s.semester = 'Fall' and s.year = 2009 and
      exists (select * from section t
              where t.semester = 'Spring' and t.year = 2010 and
                    t.course_id = s.course_id
              );
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[39]: [('CS-101',)]
```

Students taken all courses offered in the Biology department

```
In [40]: %%sql
select distinct s.ID, s.name
from student s
where not exists(
    (select course_id from course
     where dept_name = 'Biology')
    except
    (select t.course_id
     from takes t
     where t.ID = s.ID)
);
```

```
* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
0 rows affected.
```

```
Out[40]: []
```

1.4.2 Subqueries in From Clause (derived tables)

Find the average instructors' salaries of those departments where the average salary is greater than \$42,000."

```
In [41]: %%sql
        select r.dept_name, r.avg_salary
        from (
            select dept_name, floor(avg(salary)) avg_salary
            from instructor
            group by dept_name) r
        where r.avg_salary > 42000;

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[41]: [('Biology', Decimal('72000')),
          ('Comp. Sci.', Decimal('77333')),
          ('Elec. Eng.', Decimal('80000')),
          ('Finance', Decimal('85000')),
          ('History', Decimal('61000')),
          ('Physics', Decimal('91000'))]
```

WITH clause Find all departments with the maximum budget

```
In [42]: %%sql
        with max_budget (value) as(
            select max(budget)
            from department
        )
        select d.dept_name
        from department d, max_budget m
        where d.budget = m.value;

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[42]: [('Finance',)]
```

1.4.3 Scalar Subqueries in SELECT Clause

List all departments along with the number of instructors in each department

```
In [43]: %%sql
        select d.dept_name, (
            select count(*)
            from instructor i
```

```

        where i.dept_name = d.dept_name
        ) [No_Instructors]
    from department d;

```

```

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.

```

```

Out[43]: [('Biology', 1),
          ('Comp. Sci.', 3),
          ('Elec. Eng.', 1),
          ('Finance', 2),
          ('History', 2),
          ('Music', 1),
          ('Physics', 2)]

```

1.5 Aggregations & Group by

1.5.1 Aggregation functions

```

In [44]: %%sql
         select avg(salary) avg_salary
         from instructor
         where dept_name = 'Comp. Sci.';

```

```

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.

```

```

Out[44]: [(Decimal('77333.333333'),)]

```

```

In [45]: %%sql
         select count(*)
         from instructor
         where dept_name = 'Comp. Sci.';

```

```

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.

```

```

Out[45]: [(3,)]

```

```

In [46]: %%sql
         select count(dept_name) No_Dept
         from instructor;

```

```

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.

```

```
Out[46]: [(12,)]
```

```
In [47]: %%sql
        select count(distinct dept_name) No_Dept
        from instructor;

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[47]: [(7,)]
```

1.5.2 Group-by

```
In [48]: %%sql
        select dept_name, floor(avg(salary)) avg_salary
        from instructor
        group by dept_name;

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[48]: [('Biology', Decimal('72000')),
          ('Comp. Sci.', Decimal('77333')),
          ('Elec. Eng.', Decimal('80000')),
          ('Finance', Decimal('85000')),
          ('History', Decimal('61000')),
          ('Music', Decimal('40000')),
          ('Physics', Decimal('91000'))]
```

1.5.3 Having

```
In [49]: %%sql
        select dept_name, floor(avg(salary)) avg_salary
        from instructor
        group by dept_name
        having avg(salary) > 42000
        order by avg_salary desc;

* mssql+pyodbc://python:***@E7450/AP?driver=ODBC Driver 17 For SQL Server
Done.
```

```
Out[49]: [('Physics', Decimal('91000')),
          ('Finance', Decimal('85000')),
          ('Elec. Eng.', Decimal('80000')),
          ('Comp. Sci.', Decimal('77333')),
          ('Biology', Decimal('72000')),
          ('History', Decimal('61000'))]
```