

```
In [2]: import sqlite3
import pandas as pd
```

```
In [3]: con = sqlite3.connect('school.sqlite')
def x(q):
    return pd.read_sql_query(q, con)
```

```
In [4]: #available tables are buildings, class_rooms, class_sections, course_categories, course
#faculty, faculty_categories, faculty_courses, faculty_sections, student_schedules, stu
x('select * from buildings Limit 3')
```

```
Out[4]:
```

	BuildingCode	BuildingName	NumberOfFloors	ElevatorAccess	SiteParkingAvailable
0	AS	Arts and Sciences	3	True	True
1	CC	College Center	3	True	False
2	GYM	PE and Wellness	1	False	True

```
In [5]: x('select * from class_rooms Limit 3')
```

```
Out[5]:
```

	RoomNumber	BuildingCode	PhoneAvailable
0	1131	LB	True
1	1142	LB	False
2	1231	LB	True

```
In [6]: x('select * from class_sections Limit 3')
```

```
Out[6]:
```

	SectionNumber	CourseID	RoomNumber	Credits	StartTime	Duration	MondaySchedule	TuesdaySc
0	1000	11	1231	5	1899-12-30 10:00:00	50	True	
1	1002	12	1619	4	1899-12-30 15:30:00	110	True	
2	1004	13	1627	4	1899-12-30 08:00:00	50	True	



```
In [7]: x('select * from course_categories Limit 3')
```

```
Out[7]:
```

	CategoryID	Category	DepartmentID
--	------------	----------	--------------

	CategoryID	Category	DepartmentID
0	ACC	Accounting	1
1	ART	Art	3
2	BIO	Biology	2

In [8]:

```
x('select * from courses Limit 3')
```

Out[8]:

	CourseID	CategoryID	CourseCode	CourseName	CourseDescription
0	1	ACC	ACC 210	Financial Accounting Fundamentals I	Introduces basic accounting concepts, principl...
1	2	ACC	ACC 220	Financial Accounting Fundamentals II	Applications of basic accounting concepts, pri...
2	3	ACC	ACC 230	Fundamentals of Managerial Accounting	Analysis of accounting data as part of the man...

In [9]:

```
x('select * from departments Limit 3')
```

Out[9]:

	DepartmentID	DepartmentName
0	1	Business
1	2	Science & Math
2	3	Arts & Humanities

In [10]:

```
x('select * from faculty Limit 3')
```

Out[10]:

	FacultyID	FacFirstName	FacLastname	FacStreetAddress	FacCity	FacState	FacZipCode	FacAreaCoc
0	98005	Suzanne	Viescas	15127 NE 24th, #383	Redmond	WA	98052	42
1	98007	Gary	Hallmark	Route 2, Box 203B	Auburn	WA	98002	25
2	98010	Jeffrey	Smith	30301 - 166th Ave. N.E.	Fremont	CA	94538	57



In [11]:

```
x('select * from faculty_categories Limit 3')
```

Out[11]:

	FacultyID	CategoryID
0	98005	ART

	FacultyID	CategoryID
1	98005	ENG
2	98005	MUS

In [12]: `x('select * from faculty_courses Limit 3')`

Out[12]:

	FacultyID	CourseID	ProficiencyRating
0	98005	12	10
1	98005	16	10
2	98005	34	9

In [13]: `x('select * from faculty_sections Limit 3')`

Out[13]:

	SectionNumber	FacultyID
0	1000	98014
1	1002	98036
2	1004	98019

In [14]: `x('select * from student_schedules Limit 3')`

Out[14]:

	StudentID	SectionNumber	Grade
0	1001	1560	93.28
1	1001	2071	66.41
2	1001	4055	62.85

In [15]: `x('select * from students Limit 3')`

Out[15]:

	StudentID	StudFirstName	StudLastName	StudStreetAddress	StudCity	StudState	StudZipCode	Stu
0	1001	Nancy	Davolio	9877 Hacienda Drive	San Antonio	TX	78284	
1	1002	Andrew	Fuller	908 W. Capital Way	Tacoma	WA	98413	
2	1003	Sarah	Leverling	611 Alpine Drive	Palm Springs	CA	92263	



In [16]: `df=x('select avg(salary), departmentname from faculty natural join faculty_categories n
df`

Out[16]:

	avg(salary)	DepartmentName
0	54210.526316	Arts & Humanities
1	54000.000000	Business
2	49750.000000	Computer Science
3	47304.347826	Science & Math
4	51400.000000	Social Science

In [17]:

```
df=x('select avg(salary), categoryID from faculty natural join faculty_categories natur
df
```

Out[17]:

	avg(salary)	CategoryID
0	45000.000000	BIO
1	45000.000000	CHE
2	45000.000000	PHY
3	47000.000000	CSC
4	49416.666667	MAT
5	50250.000000	HIS
6	50333.333333	ECO
7	50666.666667	CIS
8	52666.666667	ACC
9	53333.333333	POL
10	53375.000000	ART
11	54000.000000	GEG
12	54571.428571	ENG
13	55250.000000	MUS
14	55500.000000	BUS
15	56000.000000	JRN

In [18]:

```
df=x('select * from student_schedules natural join students order by grade desc Limit 3
df
```

Out[18]:

	StudentID	SectionNumber	Grade	StudFirstName	StudLastName	StudStreetAddress	StudCity	Stud
0	1016	1168	98.74	Steven	Pundt	2500 Rosales Lane	Dallas	
1	1007	1012	98.36	Elizabeth	Hallmark	Route 2, Box 203B	Auburn	
2	1010	1562	98.26	Mary	Fuller	908 W. Capital Way	Tacoma	



```
In [19]: df=x('select grade from student_schedules order by grade')
df
```

```
Out[19]:
```

	Grade
0	56.36
1	58.27
2	61.1
3	62.85
4	63.56
...	...
84	96.36
85	97.39
86	98.26
87	98.36
88	98.74

89 rows × 1 columns

```
In [20]: df['Grade'] = df['Grade'].astype(float)
df['Grade'].dtype
```

```
Out[20]: dtype('float64')
```

```
In [21]: df1=df['Grade']>89.99
grade_A = df[df1]
print(grade_A.shape)
```

```
(21, 1)
```

```
In [22]: df['Grade'] = pd.to_numeric(df['Grade'],errors='coerce')
df1=df['Grade']<70.00
df1.sum()
```

```
Out[22]: 17
```

```
In [23]: df1=df['Grade']<69.99
df2 =df['Grade']>80.00
grade_C=89 - df2.sum() - df1.sum()
grade_C
```

```
Out[23]: 26
```

```
In [24]: df1=df['Grade']<79.99
```

```
df2 =df['Grade']>90.00
grade_B=89 - df2.sum() - df1.sum()
grade_B
```

Out[24]: 25

```
In [25]: x('select count(sectionnumber), buildingname from class_sections natural join class_roo
```

```
Out[25]:
```

	count(sectionnumber)	BuildingName
0	15	Arts and Sciences
1	22	College Center
2	31	Instructional Building
3	7	Library
4	1	Technology Building

```
In [26]: df=x('select * from class_sections')
Sat=df['SaturdaySchedule']=='True'
Sat.sum()
```

Out[26]: 2

```
In [27]: Mon=df['MondaySchedule']=='True'
Tue=df['TuesdaySchedule']=='True'
Wed=df['WednesdaySchedule']=='True'
Thur=df['ThursdaySchedule']=='True'
Fri=df['FridaySchedule']=='True'
Mon.sum()
```

Out[27]: 67

```
In [28]: MTWRF = Mon & Tue & Wed & Thur & Fri
MTWRF.sum()
```

Out[28]: 27

```
In [29]: MWF = Mon & Wed & Fri
MWF.sum()-MTWRF.sum()
```

Out[29]: 27

```
In [30]: x('select count(sectionnumber),departmentname from faculty_sections natural join facult
```

```
Out[30]:
```

	count(sectionnumber)	DepartmentName
0	27	Arts & Humanities

	count(sectionnumber)	DepartmentName
1	23	Arts & Humanities
2	14	Arts & Humanities
3	20	Business
4	13	Business
5	2	Business
6	18	Computer Science
7	6	Computer Science
8	8	Science & Math
9	12	Science & Math
10	39	Science & Math
11	12	Science & Math
12	21	Social Science
13	5	Social Science
14	15	Social Science
15	9	Social Science

In [31]: `x('select count(sectionnumber), facfirstname, faclastname from faculty_sections natural`

Out[31]:

	count(sectionnumber)	FacFirstName	FacLastname
0	5	Alastair	Black
1	5	Amelia	Buchanan
2	5	Michael	Davolio
3	4	Consuelo	Maynez
4	4	Gregory	Piercy
5	4	Michael	Hernandez
6	4	John	Leverling
7	4	Alaina	Hallmark
8	4	Albert	Buchanan
9	4	Ann	Fuller
10	4	Ann	Patterson
11	4	Gary	Hallmark
12	3	Caroline	Coie
13	3	David	Smith
14	3	Janet	Leverling

	count(sectionnumber)	FacFirstName	FacLastname
15	2	Allan	Davis
16	2	Ryan	Ehrlich
17	2	Katherine	Ehrlich
18	2	Joyce	Bonnicksen
19	2	David	Callahan
20	2	Laura	Callahan
21	2	James	Leverling
22	2	Suzanne	Viescas

In [32]: `x('select avg(proficiencyrating), facfirstname, faclastname from faculty_courses natura`

Out[32]:

	avg(proficiencyrating)	FacFirstName	FacLastname
0	9.307692	James	Leverling
1	9.250000	Michael	Davolio
2	9.200000	Ann	Fuller
3	9.166667	Caroline	Coie
4	9.000000	Suzanne	Viescas
5	9.000000	Ann	Patterson
6	9.000000	Consuelo	Maynez
7	9.000000	Michael	Hernandez
8	9.000000	Gary	Hallmark
9	9.000000	Joyce	Bonnicksen
10	9.000000	Alastair	Black
11	8.888889	Jeffrey	Smith
12	8.818182	Katherine	Ehrlich
13	8.800000	Gregory	Piercy
14	8.800000	Albert	Buchanan
15	8.571429	Laura	Callahan
16	8.500000	Allan	Davis

In [33]: `df=x('select avg(proficiencyrating), departmentname from faculty natural join faculty_c
df`

Out[33]:

	avg(proficiencyrating)	DepartmentName
0	9.232143	Business

	avg(proficiencyrating)	DepartmentName
1	9.000000	Social Science
2	8.965517	Computer Science
3	8.950292	Science & Math
4	8.942177	Arts & Humanities

```
In [34]: df = x('select coursecode, credits, departmentname from courses natural join class_sect
df.head()
```

```
Out[34]:
```

	CourseCode	Credits	DepartmentName
0	ACC 210	5	Business
1	ACC 210	5	Business
2	ACC 220	5	Business
3	ACC 230	5	Business
4	ACC 251	5	Business

```
In [35]: course_code = df['CourseCode'].tolist()
```

```
In [36]: level = []
for course in course_code:
    course=course[4:5]
    level.append(course)

level
df['Level']=level
```

```
In [37]: df
```

```
Out[37]:
```

	CourseCode	Credits	DepartmentName	Level
0	ACC 210	5	Business	2
1	ACC 210	5	Business	2
2	ACC 220	5	Business	2
3	ACC 230	5	Business	2
4	ACC 251	5	Business	2
...
71	PHY 201	5	Science & Math	2
72	PHY 203	5	Science & Math	2
73	POL 101	5	Social Science	1

	CourseCode	Credits	DepartmentName	Level
74	POL 101	5	Social Science	1
75	POL 213	5	Social Science	2

76 rows × 4 columns

In [38]:

```
df['Level'] = pd.to_numeric(df['Level'],errors='coerce')
df['Credits'] = pd.to_numeric(df['Credits'],errors='coerce')
df
```

Out[38]:

	CourseCode	Credits	DepartmentName	Level
0	ACC 210	5	Business	2
1	ACC 210	5	Business	2
2	ACC 220	5	Business	2
3	ACC 230	5	Business	2
4	ACC 251	5	Business	2
...
71	PHY 201	5	Science & Math	2
72	PHY 203	5	Science & Math	2
73	POL 101	5	Social Science	1
74	POL 101	5	Social Science	1
75	POL 213	5	Social Science	2

76 rows × 4 columns

In [39]:

```
groups=['Level', 'DepartmentName']
by_level = df.groupby(groups, as_index=False).mean()
by_level
```

Out[39]:

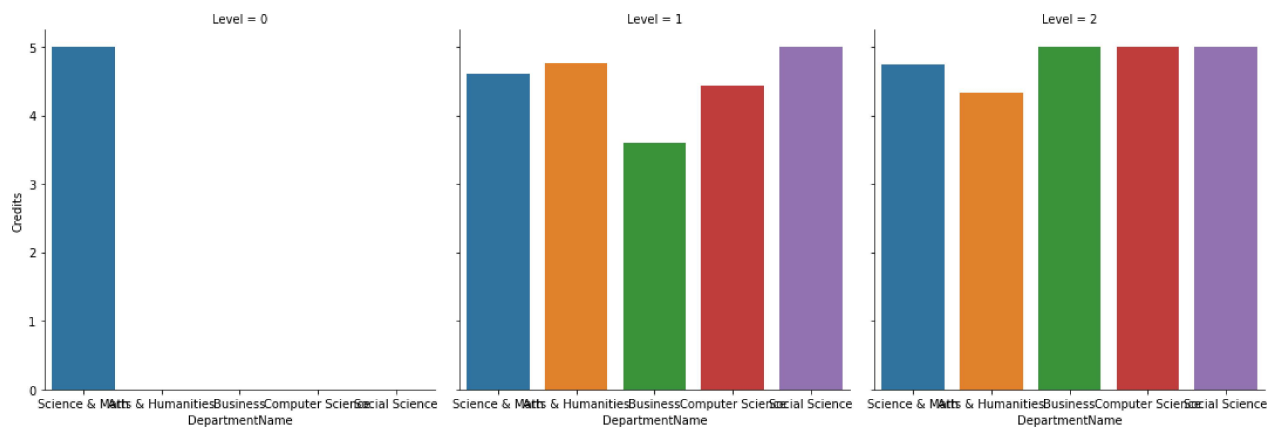
	Level	DepartmentName	Credits
0	0	Science & Math	5.000000
1	1	Arts & Humanities	4.769231
2	1	Business	3.600000
3	1	Computer Science	4.428571
4	1	Science & Math	4.600000
5	1	Social Science	5.000000
6	2	Arts & Humanities	4.333333
7	2	Business	5.000000

Level	DepartmentName	Credits
8	2 Computer Science	5.000000
9	2 Science & Math	4.750000
10	2 Social Science	5.000000

```
In [40]: import seaborn as sns
```

```
In [41]: import matplotlib.pyplot as plt
```

```
In [45]: sns.catplot(x='DepartmentName', y='Credits', col='Level', kind='bar', data=by_level)
plt.show()
```



```
In [ ]:
```