

May contain typos

species

code	species
m	maple
p	pine

trees

	tree	x	y	species	diameter	priority
0	A	10	4	m	8	71
1	B	20	4	m	10	100
2	C	30	4	p	6	30
3	D	40	4	p	8	40
4	E	50	4	m	12	99

```
import sqlite3
```

```
c = sqlite3.connect("worksheet.db")
```

```
def qry(sql):
```

```
    return pd.read_sql(sql, c)
```

```
species = qry("SELECT * FROM species")
```

```
trees = qry("SELECT * FROM trees")
```

1 `trees[trees["priority"] > 90][["x", "y"]] # convert to SQL`

X Y
1 20 4
4 50 4

What is this
in SQL?

SELECT x, y

FROM trees

WHERE priority > 90

2 `qry("SELECT x+y FROM trees WHERE species = 'm'") # convert to Pandas`

x+y
0 14
1 24
2 54

x_col = trees[trees['species'] == 'm']['x']

y_col = trees[trees['species'] == 'm']['y']

x_col + y_col // Series

3 `cd = species["code"][species["species"] == "maple"].iloc[0]`

`trees[trees["species"] == cd][["tree"]] # convert to 2 SQL queries`

(cd = 'm')

tree
0 A
1 B
4 E

SELECT code

FROM species

WHERE species = 'maple'

SELECT tree

FROM trees

WHERE species = 'm'

4 `qry("SELECT species FROM trees ORDER BY priority DESC")`

species

0 m
1 m
2 m
3 m
4 p
p

trees.sort_values("priority", ascending=False)

species

5

code	species
m	maple
p	pine

trees

tree	x	y	species	diameter	priority
A	10	4	m ^m	71 ⁷¹	71
B	20	4	m ^m	10	100
C	30	4	p	6	30
D	40	4	p	8	40
E	50	4	m	12	99

```
list(qry("SELECT tree, priority FROM trees " +
        "ORDER BY priority DESC LIMIT 1").iloc[0])
```

tree priority → tree B → ['B', 100]
0 B 100 → priority 100

```
list(trees.sort_values("priority", ascending=False).iloc[0])
```

```
6 qry("""SELECT COUNT(SPECIES) AS c1,
        COUNT(DISTINCT SPECIES) as c2
        FROM trees""")
```

c1 c2
0 5 2

```
7 qry("""SELECT species, COUNT(SPECIES) AS count,
        AVG(diameter) AS size
        FROM trees
        GROUP BY species ORDER BY count DESC""")
```

species count size
0 m 3 10
1 p 2 7

hydrants

year	color	style	owner	alt	active
1999	red	K-81	private	1179	0
2000	red	M-3	public	1065	0
2001	green	Pacer	private	1058	1
2010	blue	Pacer	public	1081	1
2014	blue	Pacer	public	1052	1
2018	blue	Pacer	public	1109	1

hydrants = qry("""
SELECT * FROM hydrants
""")

8 qry("SELECT color, year FROM hydrants WHERE color = 'blue'")

hydrants[hydrants['color'] == 'blue']
[[hydrants['year'], hydrants['color']]]

	color	year
0	blue	2010
1	blue	2014
2	blue	2018

9 df = qry("SELECT color, year FROM hydrants")

df[df.color == "blue"]

	color	year
0	red	1999
1	red	2000
2	green	2001
3	blue	2010
4	blue	2014
5	blue	2018

df["color"]

active = 1

10 qry("SELECT year FROM hydrants WHERE owner='private' AND active")

hydrants[(hydrants['owner'] == 'private') &
(hydrants['active'] == 1)]

	year
0	2001

11 df = qry("SELECT year, style, active FROM hydrants")

df[df.active == 1][["style"]]

	year	style	active
0	1999	K-81	0
1	2000	M-3	0
2	2001	Pacer	1
3	2010	Pacer	1
4	2014	Pacer	1
5	2018	Pacer	1

~~style~~

2 Pacer

3 Pacer

4 Pacer

5 Pacer

DataFrame

Series

hydrants

year	color	style	owner	alt	active
1999	red	K-81	private	1179	0
2000	red	M-3	public	1065	0
2001	green	Pacer	private	1058	1
2010	blue	Pacer	public	1081	1
2014	blue	Pacer	public	1052	1
2018	blue	Pacer	public	1109	1

hydrants = qry("""
SELECT * FROM hydrants
""")

← returns in descending order

12 hydrants["color"].value_counts() # convert to SQL

Blue 3 series SELECT color, COUNT(*) as c
red 2 FROM hydrants
green 1 GROUP BY color
ORDER BY c DESC

13 qry("""SELECT color, COUNT(*) FROM hydrants
WHERE active GROUP BY color""")

color count(*)
green 1
blue 3

14 qry("""SELECT color, COUNT(*) AS count FROM hydrants
GROUP BY color HAVING count > 1""")

color count
red 2
blue 3

15 qry("""SELECT color, COUNT(*) AS count
FROM hydrants WHERE year >= 2000
GROUP BY color HAVING count < 2""")

color count
red 1
green 1
~~blue 3~~