

SELECT ___ FROM tips; COLUMNS

*	tips
tip	tips ['tip']
tip, time	tips [['tip', 'time']]
tip AS fee	tips .rename(columns = {'tip': 'fee'}) ['fee']

SELECT ___ FROM tips; AGGREGATIONS

COUNT(*)	tips.shape[0]
AVG(tip) SUM(tip) MAX(tip) MIN(tip)	tips['tip'].mean() .sum() .max() .min()
AVG(tip), SUM(total_bill)	tips .agg({'tip': 'mean', 'total_bill': 'sum'})
AVG(tip) AS mean, SUM(tip) AS total	tips .agg(mean = ('tip', 'mean') total = ('tip', 'sum'))

SELECT * FROM tips LIMIT 10;

10	tips .head(10)
----	-------------------

SELECT ___ FROM tips GROUP BY ___;

day, AVG(tip)	tips .groupby('day') .agg({'tip': 'mean'})
day, time, COUNT(*), AVG(tip)	tips .groupby(['day', 'time']) .agg({'tip': ['size', 'mean']})

SELECT * FROM tips WHERE ___;

tip >= 1 tip = 1 tip <= 1 tip <> 1	tips [tips['tip'] >= 1] [tips['tip'] == 1] [tips['tip'] <= 1] [tips['tip'] != 1]
tip BETWEEN 1 AND 4	tips [(tips['tip'] >= 1) & (tips['tip'] <= 4)]
tip NOT BETWEEN 1 AND 4	tips [~((tips['tip'] >= 1) & (tips['tip'] <= 4))]
time LIKE '%unch' time LIKE 'Lunc%' time LIKE '%unc%	tips [tips['tip'] .str .endswith('unch') .startswith('Lunc') .contains('%unch%')]

SELECT ___ FROM tips GROUP BY ___
HAVING ___;

COUNT(*) > 10	tips .groupby('day') .filter(lambda group: len(group) > 10) .groupby('day') .agg({'tip': 'mean'})
---------------	--

CONDITIONAL COLUMNS: CASE WHEN

tips > 5 THEN 'large' ELSE 'regular' END AS tip_type	tips .assign(tip_type = tips['tip'] .apply(lambda tip: 'large' if tip > 5 else 'regular')) ['tip_type']
tips > 5 THEN 'large' WHEN tip < 1 THEN 'small' ELSE 'regular' END AS tip_type	tips .assign(tip_type = tips['tip'] .apply(lambda tip: 'large' if tip > 5 else ('small' if tip < 1 else 'regular')) ['tip_type']

CALCULATED COLUMNS: *, -, +, /

tip/total_bill AS tip_rate	tips .assign(tip_rate = tips['tip']/tips['total_bill']) ['tip_rate']
-------------------------------	---

Try statements interactively at SQL2pandas.pythonanywhere.com