Command	SQL	Pandas
SELECT	SELECT * FROM airports	airports
LIMIT	SELECT * FROM airports LIMIT 3	airports.head(3)
DISTINCT	SELECT DISTINCT type FROM airports	airports.type.unique()
WHERE	SELECT id FROM airports WHERE ident = 'KLAX'	airports [airports.ident == 'KLAX'].id
2+ Conditions	SELECT * FROM airports WHERE iso_region = 'US-CA' AND type = 'seaplane_base'	airports[(airports.iso_region == 'US-CA') & (airports.type == 'seaplane_airport')]
Specific Columns	SELECT ident, name FROM airports WHERE iso_region = 'US-CA' AND type = 'seaplane_base'	airports[(airports.iso_region == 'US-CA') & (airports.type == 'seaplane_airport')] [['ident', 'name']]
ORDER BY	SELECT * FROM airports WHERE ident = 'KLAX' ORDER BY type DESC	airports[airports.ident == 'KLAX'] .sort_values('type' , ascending=False)
IN	SELECT * FROM airports WHERE type IN ('heliport', 'balloonport')	airports[airports .type.isin(['heliport', 'balloonport'])]
NOT IN	SELECT * FROM airports WHERE type NOT IN ('heliport', 'balloonport')	airports[~airports .type.isin(['heliport', 'balloonport'])]
GROUP BY, COUNT, ORDER BY	SELECT iso_country, type, COUNT(*) FROM airports GROUP BY iso_country, type ORDER BY iso_country, type	airports.groupby(['iso_country', 'type']).size()

GROUP BY, COUNT, ORDER BY	SELECT iso_country, type, COUNT(*) FROM airports GROUP BY iso_country, type ORDER BY iso_country, COUNT(*) DESC	airports.groupby(['iso_country', 'type']).size().to_frame('size').reset_index()
		.sort_values(['iso_country', 'size'],
		ascending=[True, False])

HAVING	SELECT type, COUNT(*) FROM airports WHERE iso_country = 'US' GROUP BY type HAVING COUNT(*) > 1000	airports[airports.iso_country == 'US']
		.groupby('type').filter(lambda g: len(g) > 1000)
		.groupby('type').size()

Top-N Values	SELECT name FROM airports ORDER BY elevation_ft DESC LIMIT 10	airports.nlargest(10, columns='elevation_ft') .name
Top-N Values Offset	SELECT name FROM airports ORDER BY elevation_ft DESC LIMIT 10 OFFSET 5	airports.nlargest(10, columns='elevation_ft') .name.iloc[5:]
Aggregate Functions	SELECT MAX(elevation_ft), AVG(elevation_ft), MEDIAN(elevation_ft) FROM airports	airports.agg({' elevation_ft': ['max', 'mean', 'median']}).T

UNION ALL	SELECT name FROM airports WHERE ident = 'KLAX'
	UNION ALL
	SELECT name FROM airports WHERE ident = 'KLGB'
		pd.concat([
		airports[airports.ident == 'KLAX'] [['name']],
		airports [airports.ident == 'KLGB'] [['name']]])

UNION	SELECT name FROM airports WHERE ident = 'KLAX'
	UNION
	SELECT name FROM airports WHERE ident = 'KLGB'
		pd.concat([
		airports[airports.ident == 'KLAX'] [['name']],
		airports [airports.ident == 'KLGB'] [['name']]])
		.drop_duplicates()