|  |  |  |
| --- | --- | --- |
| **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()** |