

FROM

SQL

TO

PANDAS



@uzwalgoudvaddeboina

Select all columns



Input table df

CustID	Name
1	Doe
2	Jo
3	Tod

```
SELECT *  
FROM df;
```

Output

CustID	Name
1	Doe
2	Jo
3	Tod



```
import pandas as pd
```

```
data = {  
    'CustID': [1, 2, 3],  
    'Name': ['Doe', 'Jo', 'Tod']  
}
```

```
df = pd.DataFrame(data)
```

```
print(df)
```

output

	CustID	Name
0	1	Doe
1	2	Jo
2	3	Tod

Select specific column



```
CREATE TABLE "df" (  
  "CustID" INTEGER,  
  "Name"    VARCHAR(10)  
);  
  
INSERT INTO "df" VALUES  
(1, 'Doe'),  
(2, 'Jo'),  
(3, 'Tod')  
;
```

```
SELECT "Name"  
FROM "df"
```

Name
Doe
Jo
Tod



```
import pandas as pd  
  
data = {  
    'CustID': [1, 2, 3],  
    'Name': ['Doe', 'Jo', 'Tod']  
}  
  
df = pd.DataFrame(data)
```

```
print(df['Name'])
```

```
0    Doe  
1     Jo  
2    Tod  
Name: Name, dtype: object
```

Select specific columns



```
CREATE TABLE "df" (  
  "CustID" INTEGER,  
  "FirstName" VARCHAR,  
  "LastName" VARCHAR  
);  
  
INSERT INTO "df" VALUES  
(1, 'Doe', 'Pala'),  
(2, 'Jo', 'Noice'),  
(3, 'Tod', 'Palle')  
;  
  
SELECT  
  "CustID",  
  "FirstName"  
FROM "df";|
```

...	CustID	FirstName
	1	Doe
	2	Jo
	3	Tod



```
import pandas as pd  
  
df = pd.DataFrame(  
    columns = [  
        'CustID',  
        'FirstName',  
        'LastName'  
    ]  
)  
  
df['CustID'] = [1, 2, 3]  
  
df['FirstName'] = ['Doe', 'Jo', 'Tod']  
  
df['LastName'] = ['Pala', 'Noice', 'Palle']  
  
print(df)
```

```
   CustID  FirstName  LastName  
0        1        Doe        Pala  
1        2         Jo        Noice  
2        3         Tod        Palle
```

```
print(df[['CustID', 'FirstName']])
```

```
   CustID  FirstName  
0        1        Doe  
1        2         Jo  
2        3         Tod
```

Filter Rows



```
CREATE TABLE "df" (  
  "CustID" INTEGER,  
  "Name"    VARCHAR(10)  
);
```

```
INSERT INTO "df" VALUES  
(1, 'Doe'),  
(2, 'Jo'),  
(3, 'Tod')  
;
```

```
SELECT *  
  FROM "df"  
 WHERE "CustID" = '2';
```

CustID	Name	...
2	Jo	



```
import pandas as pd  
  
df = pd.DataFrame(  
    columns = ['CustID', 'Name']  
)  
  
df['CustID'] = [1, 2, 3]  
  
df['Name'] = ['Doe', 'Jo', 'Tod']
```

```
df[df['CustID'] == 2]
```

	CustID	Name
1	2	Jo

"From SQL to Pandas" By **Uzwal**

Limit vs Head



```
CREATE TABLE "df" (  
  "CustID" INTEGER,  
  "Name"    VARCHAR(10)  
);  
  
INSERT INTO "df" VALUES  
(1, 'Doe'),  
(2, 'Jo'),  
(3, 'Tod')  
;
```

```
SELECT *  
  FROM "df"  
 LIMIT 1;
```

...	CustID	Name
	1	Doe



```
import pandas as pd  
  
df = pd.DataFrame(  
    columns = ['CustID', 'Name']  
)  
  
df['CustID'] = [1, 2, 3]  
  
df['Name'] = ['Doe', 'Jo', 'Tod']
```

```
print(df.head(1))
```

```
   CustID  Name  
0        1  Doe
```

Distinct vs Unique



```
CREATE TABLE "df" (  
  "CustID" INTEGER,  
  "Name" VARCHAR  
);
```

```
INSERT INTO "df" VALUES  
(1, 'Doe'),  
(2, 'Jo'),  
(1, 'Tod')  
;
```

```
SELECT DISTINCT "CustID"  
FROM "df";
```

CustID
1
2



```
import pandas as pd  
  
df = pd.DataFrame(  
    columns = ['CustID', 'Name']  
)  
  
df['CustID'] = [1, 2, 1]  
  
df['Name'] = ['Doe', 'Jo', 'Tod']  
  
print(df)
```

```
   CustID Name  
0        1  Doe  
1        2   Jo  
2        1  Tod
```

```
print(df.CustID.unique())
```

```
[1 2]
```

"From SQL to Pandas" By Uzwal

distinct vs nunique



```
create table "df" (  
  "CustID" INTEGER  
);  
  
INSERT INTO df values  
(10),  
(20),  
(10);  
  
SELECT COUNT(DISTINCT "CustID")  
FROM df;
```

...	COUNT(DISTINCT "CUSTID")
	2



```
import pandas as pd  
  
df = pd.DataFrame(  
    columns = ['CustID']  
)  
  
df['CustID'] = [10, 20, 10]  
  
print(df)
```

```
   CustID  
0      10  
1      20  
2      10
```

```
print(df.CustID.nunique())
```

```
2
```


Total elements in Table/DataFrame

table/dataframe

CustID	Name
10	Doe
20	Jo
30	Tod



```
SELECT COUNT(*) * (
  SELECT COUNT(*)
  FROM INFORMATION_SCHEMA.columns
  WHERE TABLE_CATALOG = 'DATABASE_NAME'
  AND TABLE_SCHEMA = 'SCHEMA_NAME'
  AND TABLE_NAME='df'
) AS "Size"
from "df";
```

Size
6



```
df.size
```

6

Get column names, data types, etc

table/dataframe

CustID	Name
10	Doe
20	Jo
30	Tod



```
desc table "df";
```

name	...	type	kind
CustID		NUMBER(38,0)	COLUMN
Name		VARCHAR(20)	COLUMN



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 2 columns):
#   Column  Non-Null Count  Dtype
---  -
0   CustID   3 non-null        int64
1   Name     3 non-null        object
dtypes: int64(1), object(1)
memory usage: 176.0+ bytes
```

"From SQL to Pandas" By Uzwal

Descriptive Stats: Pandas | SQL



```
In [17]: df
Out[17]: 0    1
         1    2
         2    3
         3    4
         4    5
         Name: AGE, dtype: int64
```

```
In [21]: df.describe()
Out[21]: count    5.00000
         mean     3.00000
         std     1.58111
         min     1.00000
         25%     2.00000
         50%     3.00000
         75%     4.00000
         max     5.00000
         Name: AGE, dtype: float64
```



AGE
1
2
3
4
5

```
SELECT
    COUNT(age) AS "count"
,   AVG(age) AS "mean"
,   STDDEV(age) as "std"
,   MIN(age) as "min"
,   PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY age) "25%"
,   PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY age) "50%"
,   PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY age) "75%"
,   MAX(age) as "max"
FROM desc_stats;
```

...	count	mean	std	min	25%	50%	75%	max
	5	3.000000	1.58113883	1	2.000	3.000	4.000	5

GROUP BY



```
CREATE TABLE "df" (  
  "Gender" VARCHAR(1)  
, "Population" INTEGER  
);  
  
INSERT INTO "df" VALUES  
( 'M', 1),  
( 'F', 1),  
( 'M', 0),  
( 'F', 1)  
;  
  
SELECT  
  "Gender"  
, SUM("Population")  
FROM "df"  
GROUP BY "Gender"  
;
```

Gender	Population
M	1
F	2



```
: import pandas as pd  
  
df = {  
    'Gender': ['M', 'F', 'M', 'F'],  
    'Population': [1, 1, 0, 1]  
}  
  
df = pd.DataFrame(df)  
  
df
```

```
:  
  
      Gender  Population  
0         M            1  
1         F            1  
2         M            0  
3         F            1
```

```
: print(df.groupby('Gender').sum());  
  
      Population  
Gender  
F             2  
M             1
```



Sort by Column



```
create or replace table "df" (  
  "ID" INTEGER,  
  "Name" VARCHAR(10)  
);
```

```
INSERT INTO "df" values  
(5, 'Joe'),  
(2, 'Doe'),  
(4, 'Paula'),  
(3, 'John'),  
(1, 'Terry')  
;
```

```
SELECT *  
FROM "df"  
ORDER BY "ID";
```

...	ID	Name
	1	Terry
	2	Doe
	3	John
	4	Paula
	5	Joe



```
import pandas as pd
```

```
df = {  
    'ID': [5, 2, 4, 3, 1],  
    'NAME': ['Joe', 'Doe', 'Paula', 'John', 'Terry']  
}
```

```
df = pd.DataFrame(df)
```

```
df.sort_values(by=['ID'])
```

	ID	NAME
4	1	Terry
1	2	Doe
3	3	John
2	4	Paula
0	5	Joe



Sort by Multiple Columns



```
create or replace table "df" (  
  "ID"      INTEGER,  
  "Name"    VARCHAR(10),  
  "AGE"     INTEGER  
);
```

```
INSERT INTO "df" values  
(5, 'Joe', 20),  
(2, 'Doe', 50),  
(2, 'Paula', 10),  
(1, 'John', 40),  
(1, 'Terry', 30)  
;
```

```
SELECT *  
FROM "df"  
ORDER BY "ID", "AGE";
```

...	ID	Name	AGE
	1	Terry	30
	1	John	40
	2	Paula	10
	2	Doe	50
	5	Joe	20



```
import pandas as pd  
  
df = {  
    'ID': [5, 2, 2, 1, 1],  
    'NAME': ['Joe', 'Doe', 'Paula', 'John', 'Terry'],  
    'AGE': [20, 50, 10, 40, 30]  
}
```

```
df = pd.DataFrame(df)  
  
df.sort_values(by=['ID', 'AGE'])
```

	ID	NAME	AGE
4	1	Terry	30
3	1	John	40
2	2	Paula	10
1	2	Doe	50
0	5	Joe	20

**Sorted by default
in ascending order**

Value Count



```
create or replace table "df" (  
  "NAME"    VARCHAR(10)  
);
```

```
INSERT INTO "df" values  
( 'Joe' ),  
( 'Doe' ),  
( 'Paula' ),  
( 'Joe' ),  
( 'Doe' )  
;
```

```
SELECT "NAME", COUNT(*)  
FROM "df"  
GROUP BY "NAME"  
ORDER BY COUNT(*) DESC;
```

NAME	... COUNT(*)
Joe	2
Doe	2
Paula	1



```
import pandas as pd
```

```
df = ['Joe', 'Doe', 'Paula', 'Joe', 'Doe']
```

```
df = pd.DataFrame(df)
```

```
df.value_counts()
```

```
Doe      2  
Joe      2  
Paula    1  
Name: count, dtype: int64
```

Drop Duplicates - If All Columns Duplicated



```
create or replace table "df" (  
  "ID"      INTEGER,  
  "NAME"    VARCHAR(10)  
);
```

```
INSERT INTO "df" values  
(1, 'Joe'),  
(2, 'Jack'),  
(3, 'Paula'),  
(1, 'Joe')  
;
```

```
SELECT DISTINCT *  
FROM "df"  
;
```

...	ID	NAME
	1	Joe
	2	Jack
	3	Paula



```
: import pandas as pd
```

```
: df = {  
    'ID': [1, 2, 3, 1],  
    'Name': ['Joe', 'Jack', 'Paul', 'Joe']  
}
```

```
: df = pd.DataFrame(df)
```

```
: df
```

	ID	Name
0	1	Joe
1	2	Jack
2	3	Paul
3	1	Joe

```
df.drop_duplicates()
```

	ID	Name
0	1	Joe
1	2	Jack
2	3	Paul



Uzwal Goud Vaddeboina

Found this post
helpful?

follow



Uzwal for tips on
SQL, Python,
and Data Analytics.