FROM SQL TO PANDAS

Uzwal Goud Vaddeboina

INDEX

SELECT

SELECT ALL COLUMNS
SELECT SINGLE COLUMN
SELECT MULTIPLE COLUMNS

WHERE

EQUAL TO (=)
NOT EQUAL TO (!=)
GREATER THAN (>)
GREATER THAN EQUAL TO (>=)
LESS THAN (<)
LESS THAN EQUAL TO (<=)
AND
OR
IN
NOT IN

ORDER BY

SORT BY SINGLE COLUMN ASC
SORT BY SINGLE COLUMN DESC
SORT BY MULTIPLE COLUMNS ASC
SORT BY MULTIPLE COLUMNS DESC
SORT BY ASC AND DESC

LIMIT & OFFSET

TOP N ROWS
OFFSET

GROUP

GROUP BY SINGLE COLUMN
GROUP BY MULTIPLE COLUMNS

JOINS

INNER JOIN
LEFT JOIN
RIGHT JOIN
FULL JOIN
CROSS JOIN

UNION & UNION ALL

UNION BY SINGLE COLUMN
UNION BY ALL COLUMNS
UNION ALL BY SINGLE COLUMN
UNION ALL BY ALL COLUMNS

INSERT

ADD SINGLE COLUMN

UPDATE

UPDATE SINGLE COLUMN SINGLE ROW
UPDATE MULTIPLE COLUMNS SINGLE ROW
UPDATE SINGLE COLUMN MULTIPLE ROWS
UPDATE MULTIPLE COLUMNS MULTIPLE ROWS
UPDATE SINGLE COLUMN ALL ROWS
UPDATE MULTIPLE COLUMNS ALL ROWS

Uzwal Goud Vaddeboina

INDEX

DROP

DROP SINGLE COLUMN

DROP MULTIPLE COLUMNS

RENAME

RENAME SINGLE COLUMN

RENAME MULTIPLE COLUMNS

COUNT

COUNT OF DISTINCT VALUES

COUNT OF TOTAL VALUES

COUNT OF UNIQUE VALUES

AGGREGATIONS

DESCRIPTIVE STATISTICS

DUPLICATES

DISTINCT VALUES

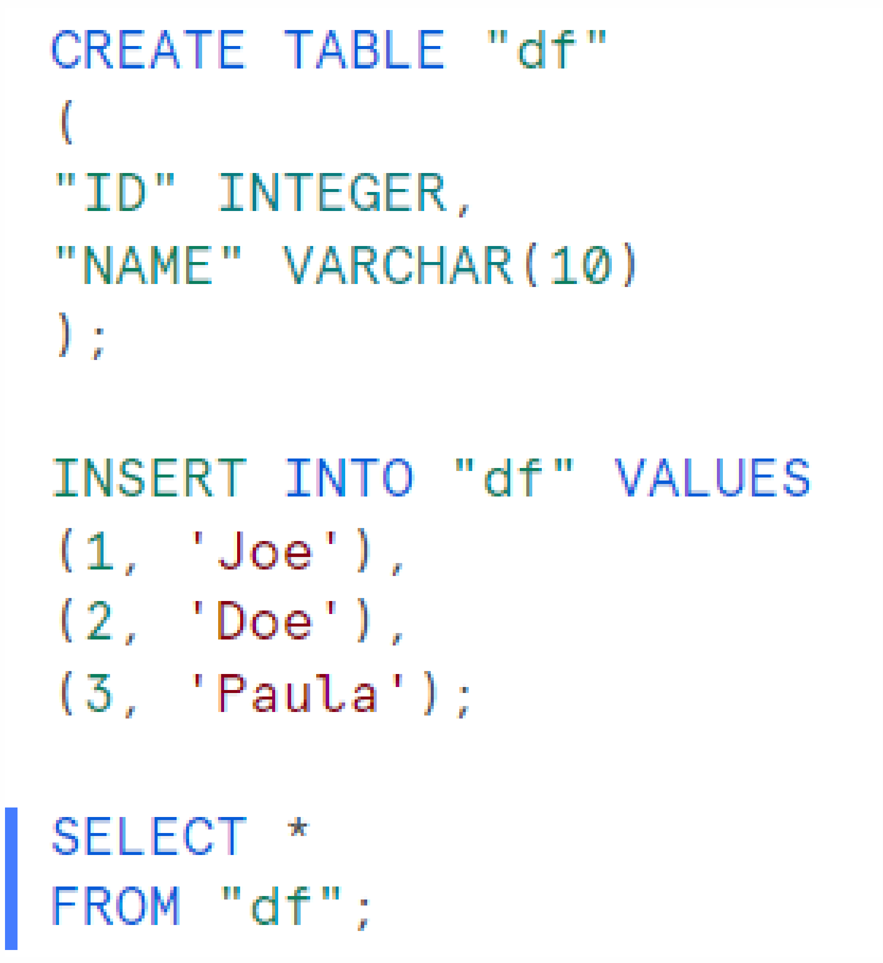
DROP ROW - ALL COLUMNS DUPLICATED

DROP ROW - KEY COLUMN DUPLICATED

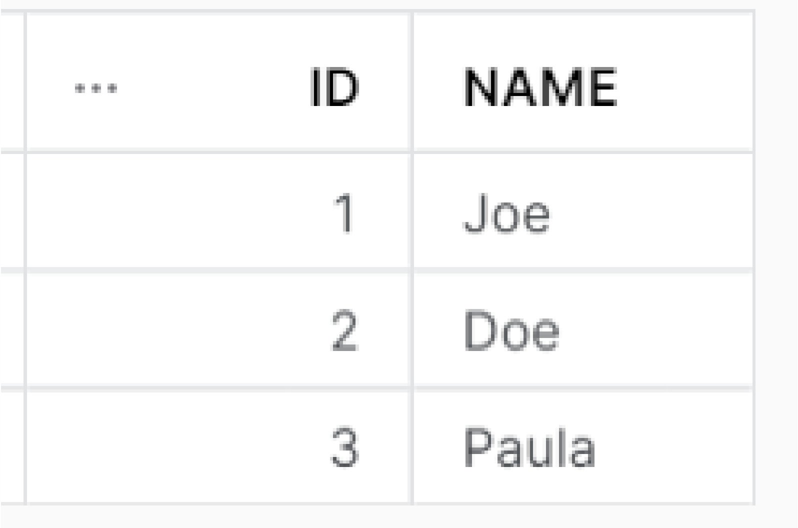
DDL

STRUCTURE OF TABLE

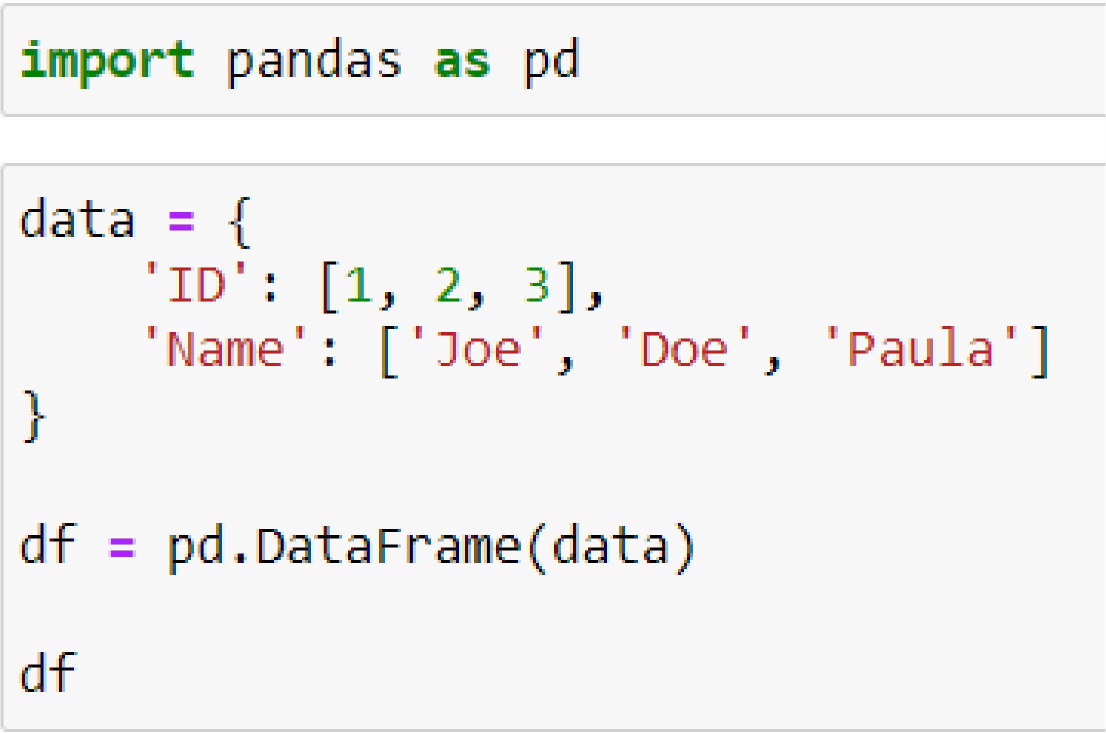
SELECT ALL COLUMNS



```
CREATE TABLE "df"
(
"ID" INTEGER,
"NAME" VARCHAR(10)
);

INSERT INTO "df" VALUES
(1, 'Joe'),
(2, 'Doe'),
(3, 'Paula');

SELECT *
FROM "df";
```

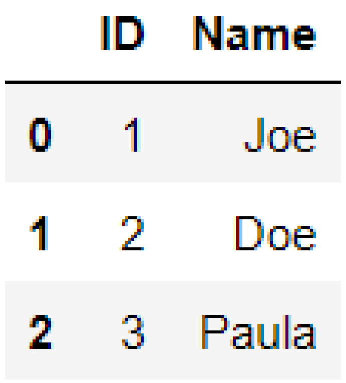


...	ID	NAME
	1	Joe
	2	Doe
	3	Paula



```
import pandas as pd
```

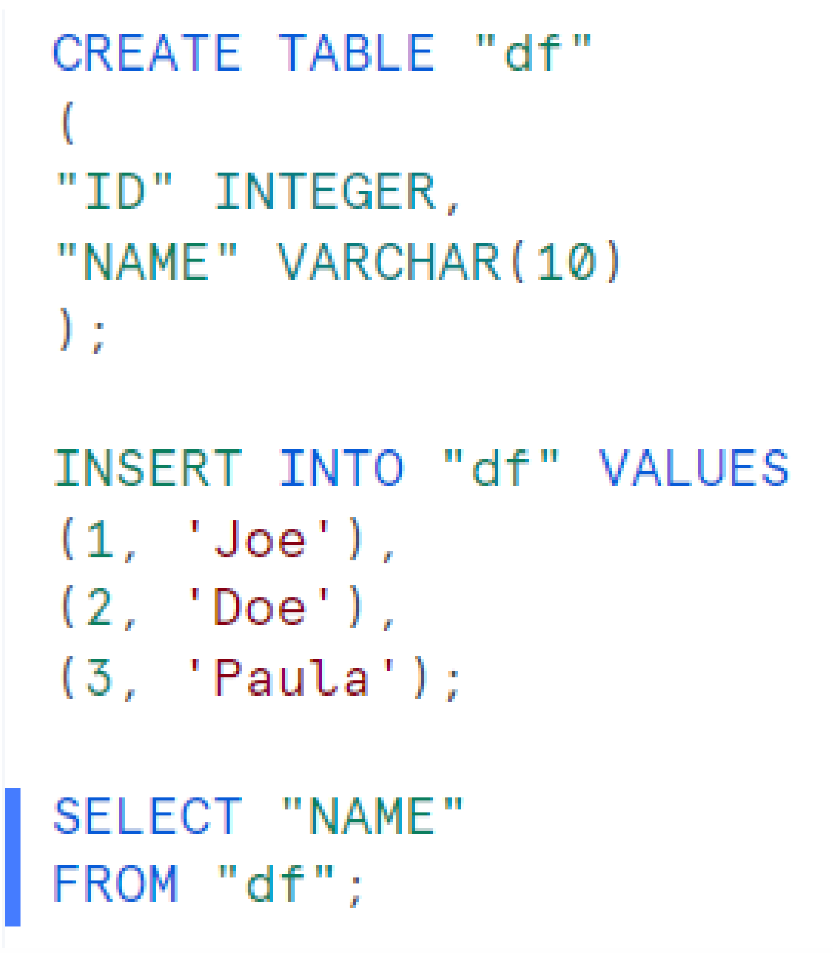
```
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Doe', 'Paula']
}

df = pd.DataFrame(data)

df
```



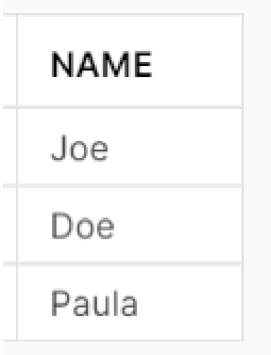
	ID	Name
0	1	Joe
1	2	Doe
2	3	Paula

in Uzwal Goud Vaddeboina

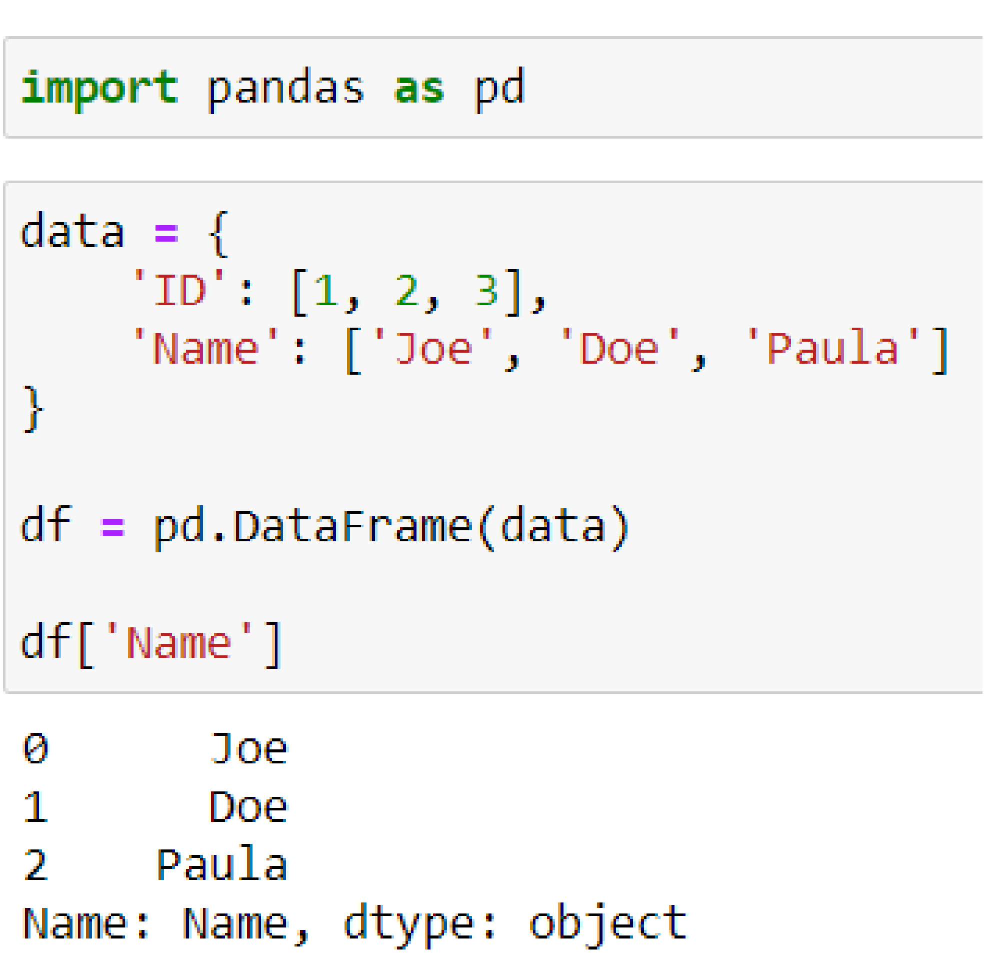
SELECT SINGLE COLUMN



```sql
CREATE TABLE "df"
(
"ID" INTEGER,
"NAME" VARCHAR(10)
);

INSERT INTO "df" VALUES
(1, 'Joe'),
(2, 'Doe'),
(3, 'Paula');

SELECT "NAME"
FROM "df";
```



NAME
Joe
Doe
Paula



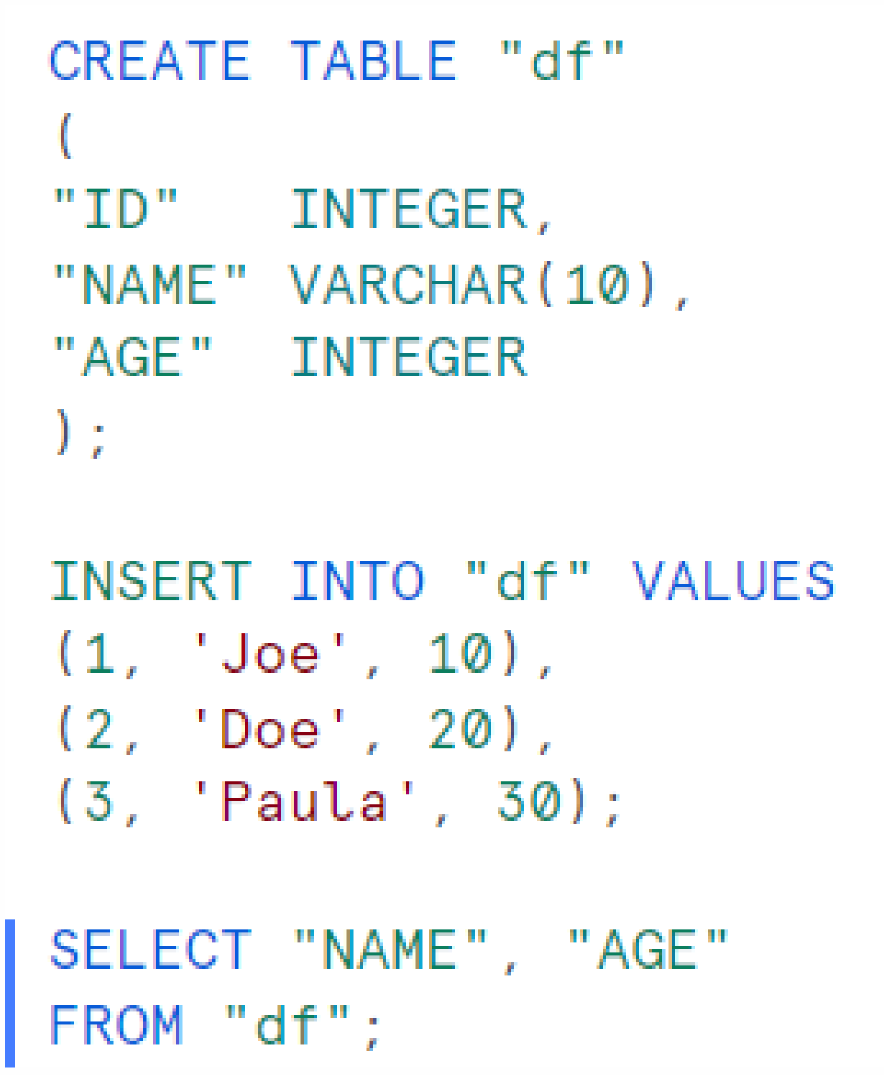
```python
import pandas as pd
```

```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Doe', 'Paula']
}

df = pd.DataFrame(data)

df['Name']
```

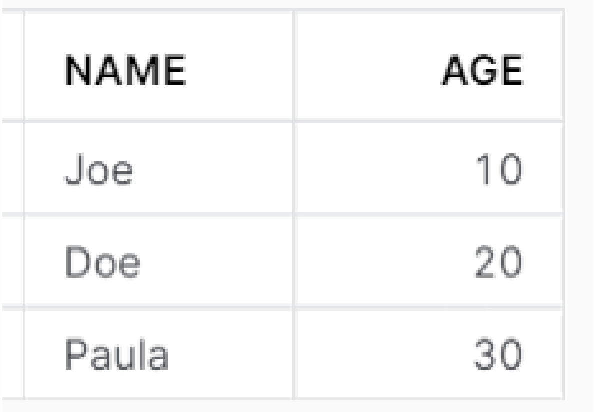
```
0      Joe
1      Doe
2    Paula
Name: Name, dtype: object
```

in Uzwal Goud Vaddeboina

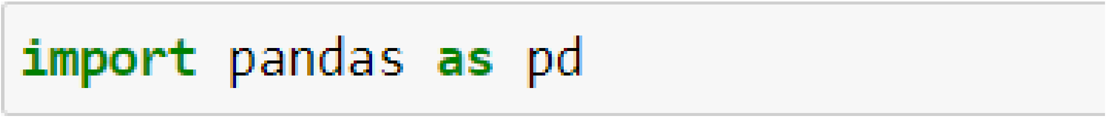
SELECT MULTIPLE COLUMNS



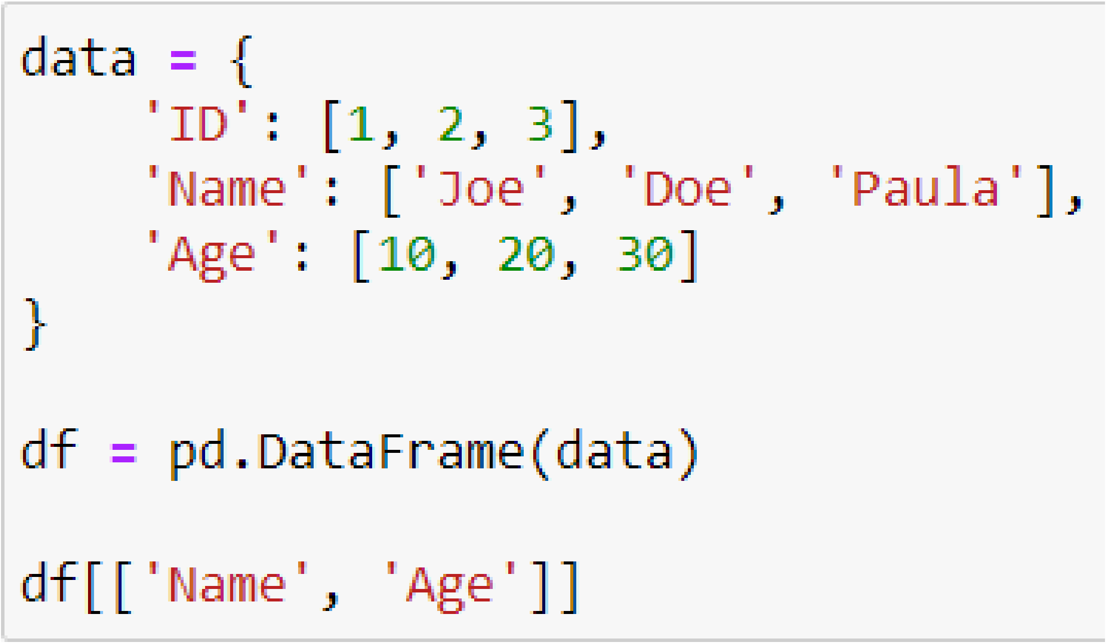
```sql
CREATE TABLE "df"
(
"ID"    INTEGER,
"NAME"  VARCHAR(10),
"AGE"   INTEGER
);

INSERT INTO "df" VALUES
(1, 'Joe', 10),
(2, 'Doe', 20),
(3, 'Paula', 30);

SELECT "NAME", "AGE"
FROM "df";
```



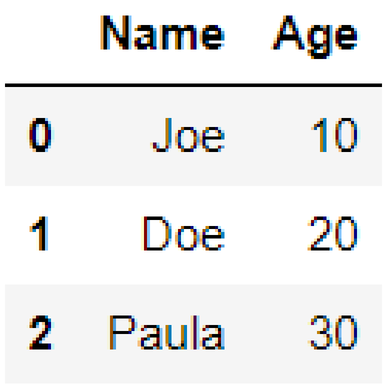
NAME	AGE
Joe	10
Doe	20
Paula	30



```python
import pandas as pd
```



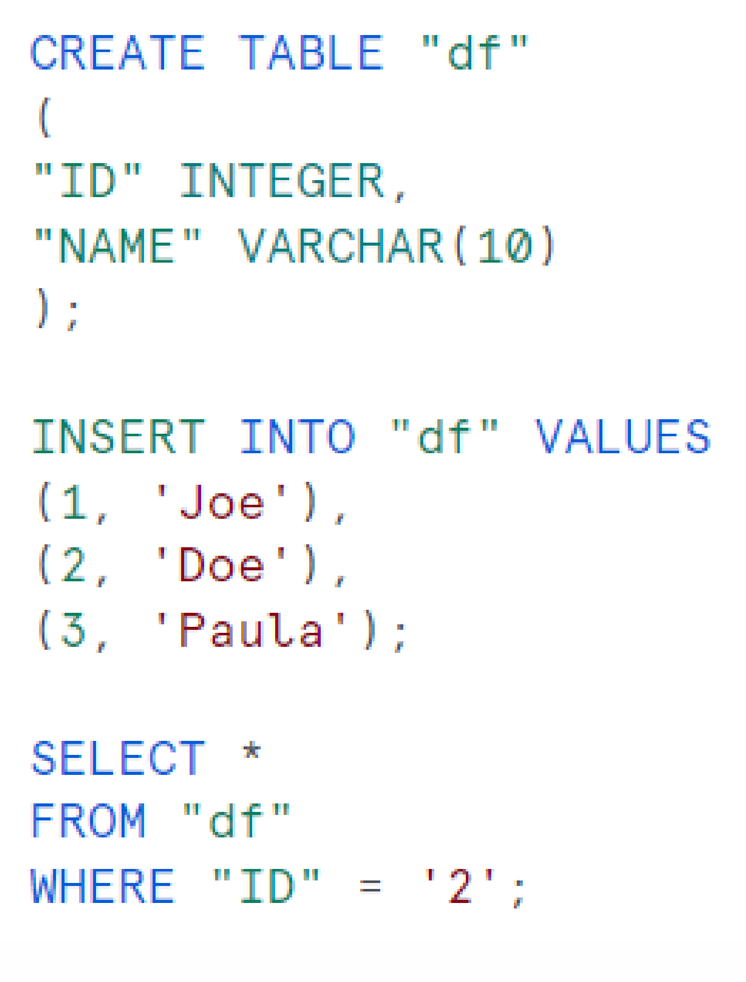
```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Doe', 'Paula'],
    'Age': [10, 20, 30]
}

df = pd.DataFrame(data)

df[['Name', 'Age']]
```



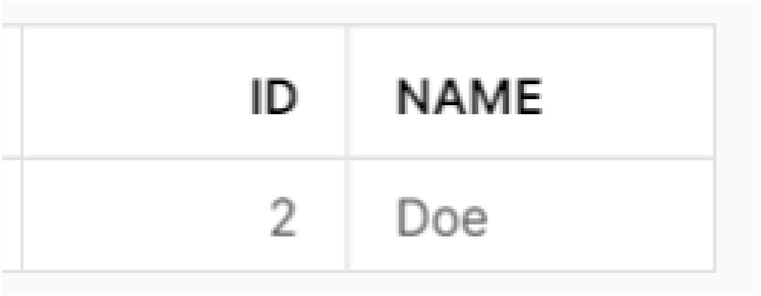
	Name	Age
0	Joe	10
1	Doe	20
2	Paula	30

in Uzwal Goud Vaddeboina

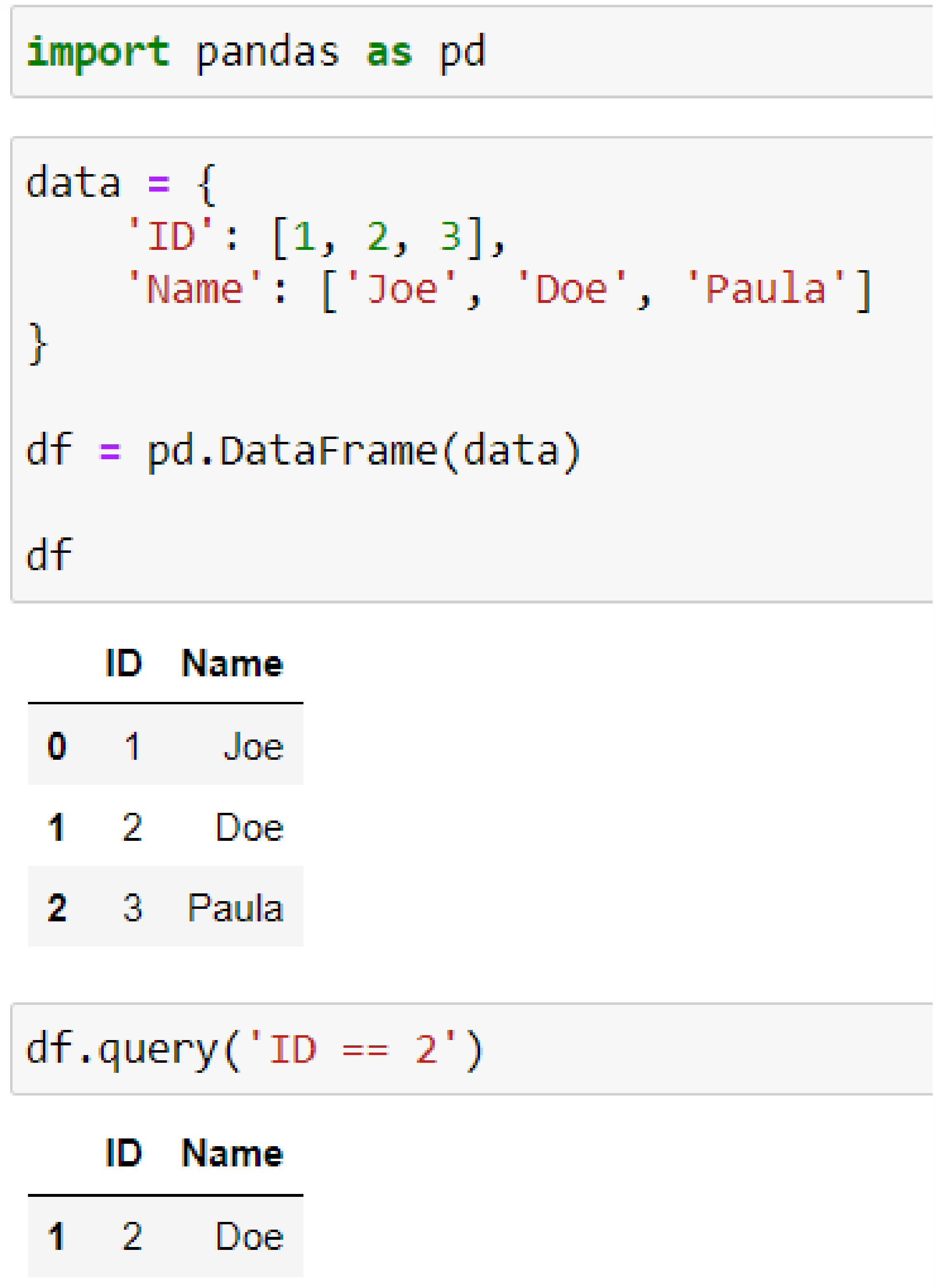
EQUAL TO (=)



```sql
CREATE TABLE "df"
(
"ID" INTEGER,
"NAME" VARCHAR(10)
);

INSERT INTO "df" VALUES
(1, 'Joe'),
(2, 'Doe'),
(3, 'Paula');

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



ID	NAME
2	Doe



```python
import pandas as pd
```

```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Doe', 'Paula']
}

df = pd.DataFrame(data)

df
```

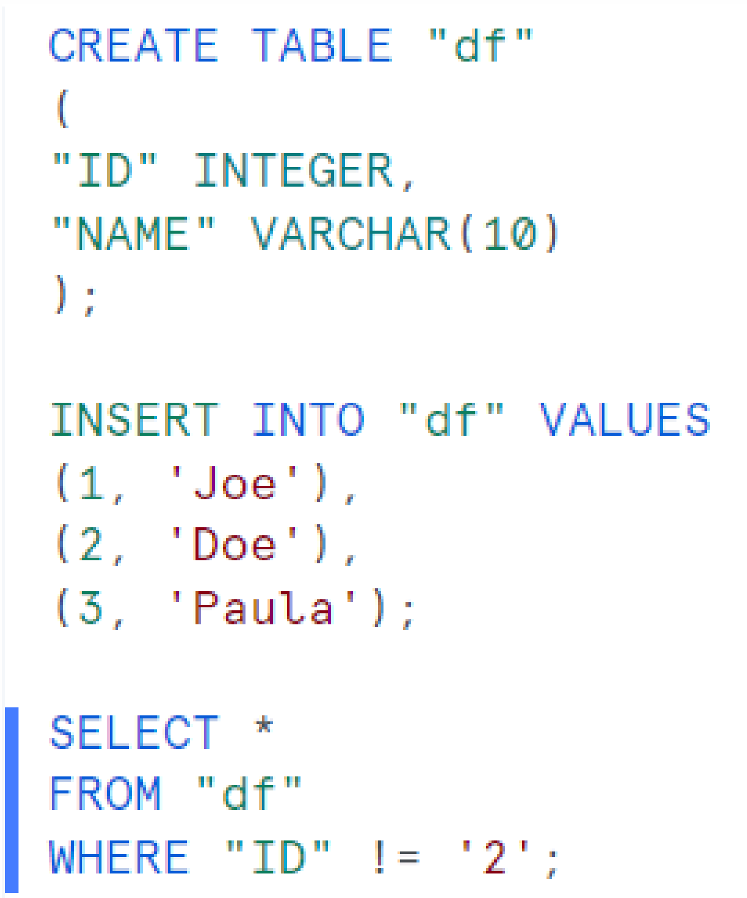
	ID	Name
0	1	Joe
1	2	Doe
2	3	Paula

```python
df.query('ID == 2')
```

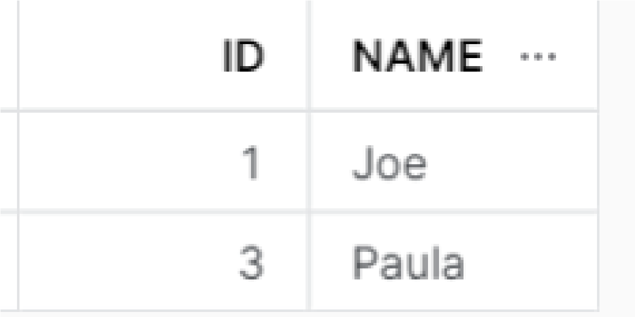
	ID	Name
1	2	Doe

Uzwal Goud Vaddeboina

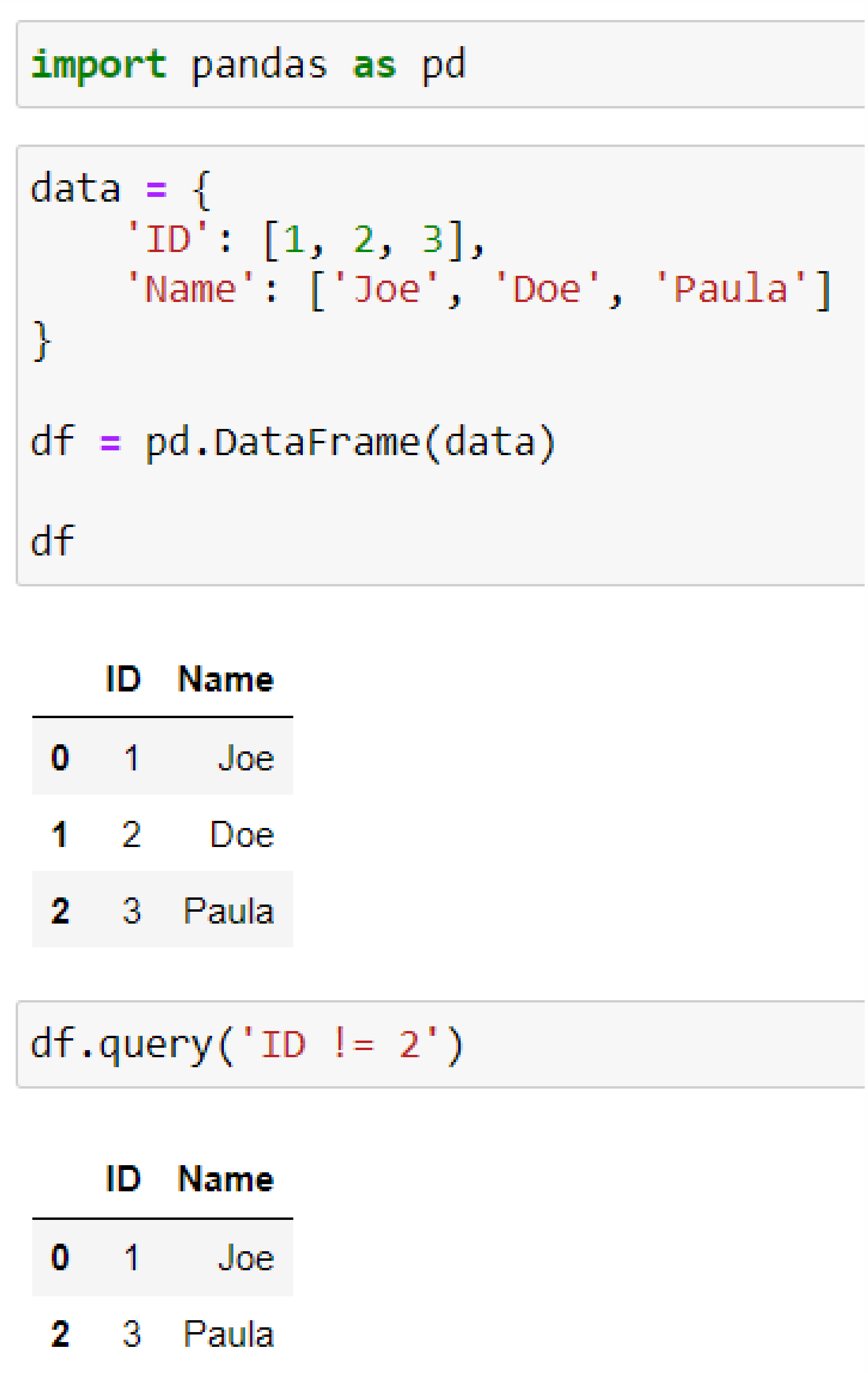
NOT EQUAL TO (!=)



```sql
CREATE TABLE "df"
(
"ID" INTEGER,
"NAME" VARCHAR(10)
);

INSERT INTO "df" VALUES
(1, 'Joe'),
(2, 'Doe'),
(3, 'Paula');

SELECT *
FROM "df"
WHERE "ID" != '2';
```



ID	NAME
1	Joe
3	Paula



```python
import pandas as pd
```

```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Doe', 'Paula']
}

df = pd.DataFrame(data)

df
```

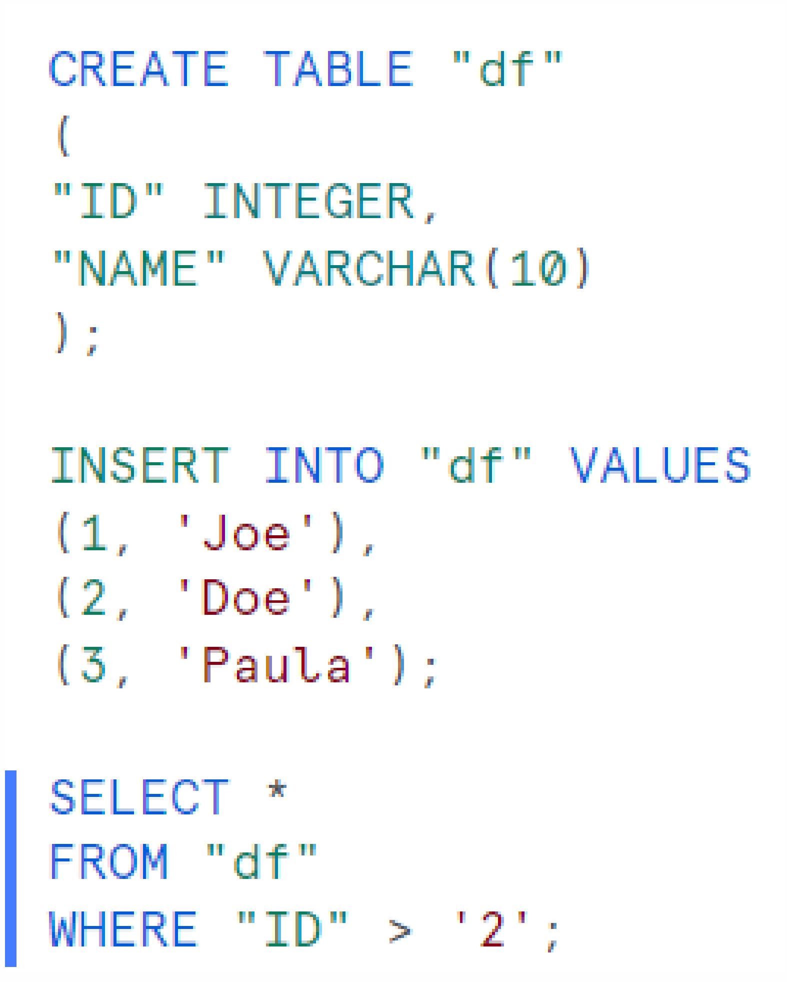
	ID	Name
0	1	Joe
1	2	Doe
2	3	Paula

```python
df.query('ID != 2')
```

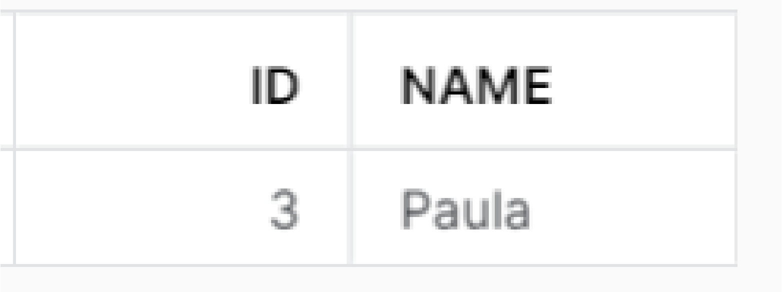
	ID	Name
0	1	Joe
2	3	Paula

Uzwal Goud Vaddeboina

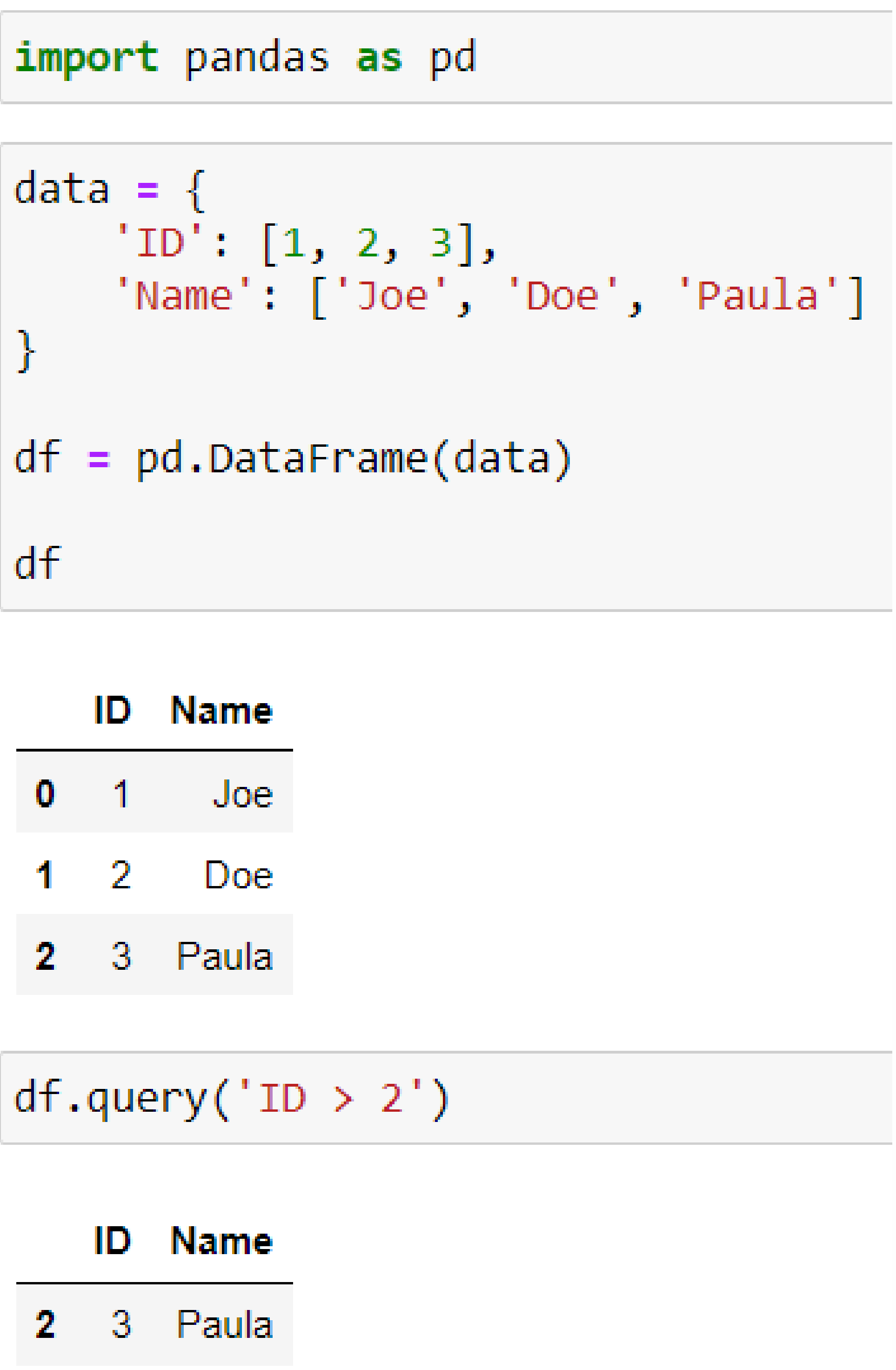
GREATER THAN (>)



```sql
CREATE TABLE "df"
(
"ID" INTEGER,
"NAME" VARCHAR(10)
);

INSERT INTO "df" VALUES
(1, 'Joe'),
(2, 'Doe'),
(3, 'Paula');

SELECT *
FROM "df"
WHERE "ID" > '2';
```



ID	NAME
3	Paula



```python
import pandas as pd
```

```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Doe', 'Paula']
}

df = pd.DataFrame(data)

df
```

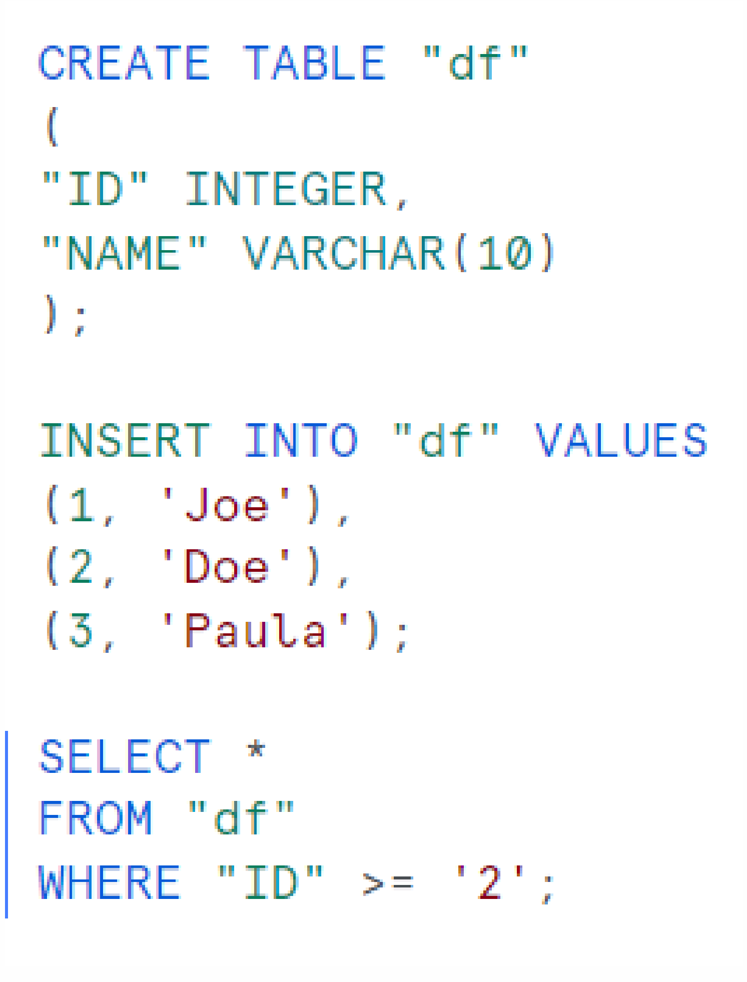
	ID	Name
0	1	Joe
1	2	Doe
2	3	Paula

```python
df.query('ID > 2')
```

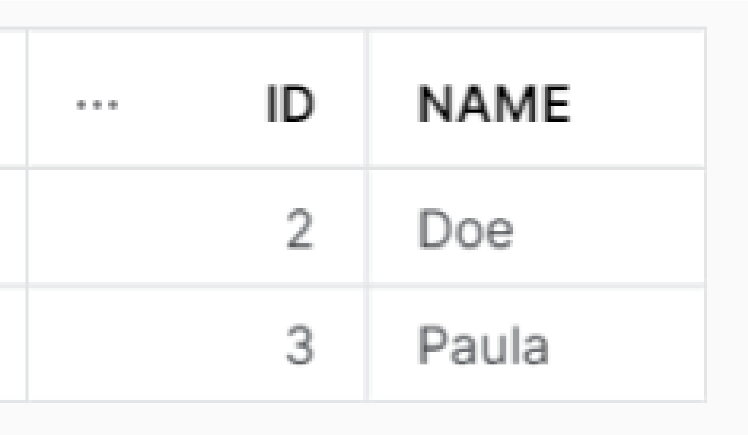
	ID	Name
2	3	Paula

Uzwal Goud Vaddeboina

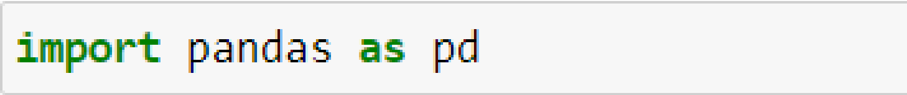
GREATER THAN EQUAL TO (>=)



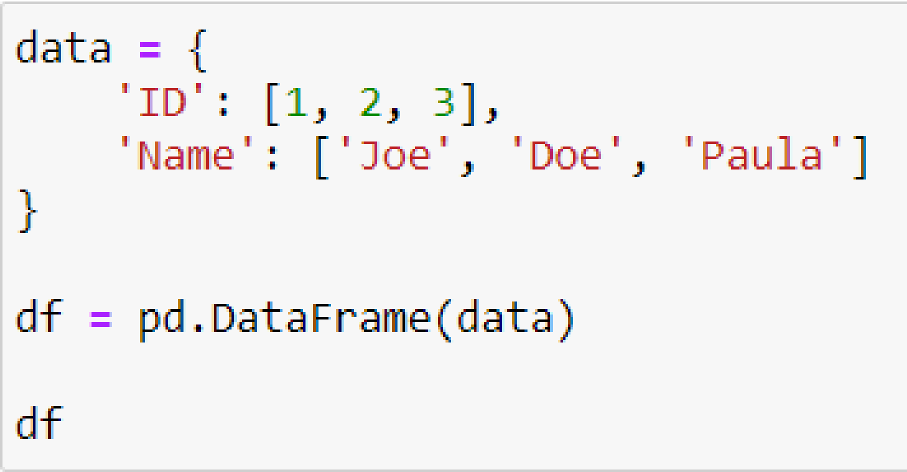
```sql
CREATE TABLE "df"
(
"ID" INTEGER,
"NAME" VARCHAR(10)
);

INSERT INTO "df" VALUES
(1, 'Joe'),
(2, 'Doe'),
(3, 'Paula');

SELECT *
FROM "df"
WHERE "ID" >= '2';
```



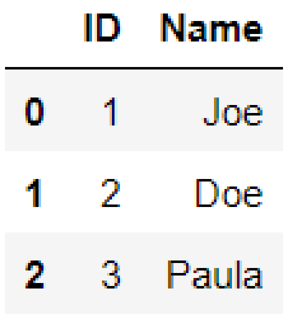
...	ID	NAME
	2	Doe
	3	Paula



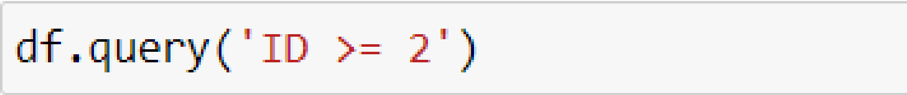
```python
import pandas as pd
```



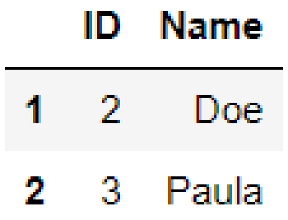
```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Doe', 'Paula']
}

df = pd.DataFrame(data)

df
```



	ID	Name
0	1	Joe
1	2	Doe
2	3	Paula



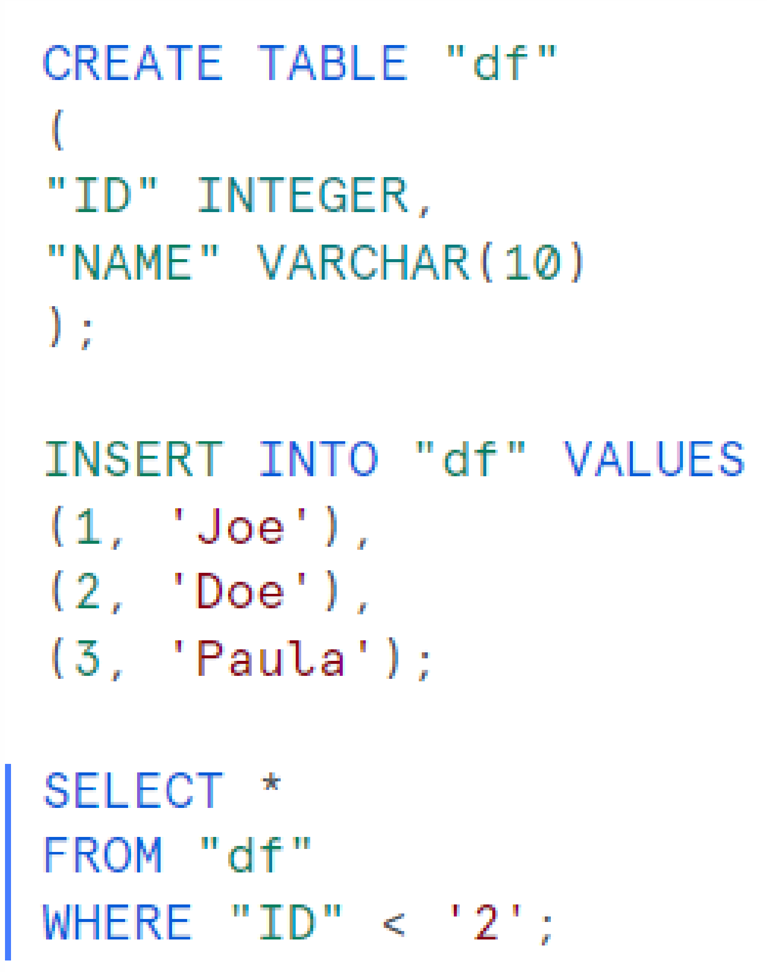
```python
df.query('ID >= 2')
```



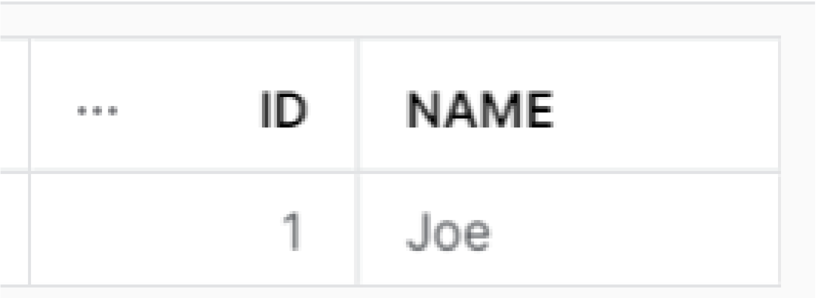
	ID	Name
1	2	Doe
2	3	Paula

Uzwal Goud Vaddeboina

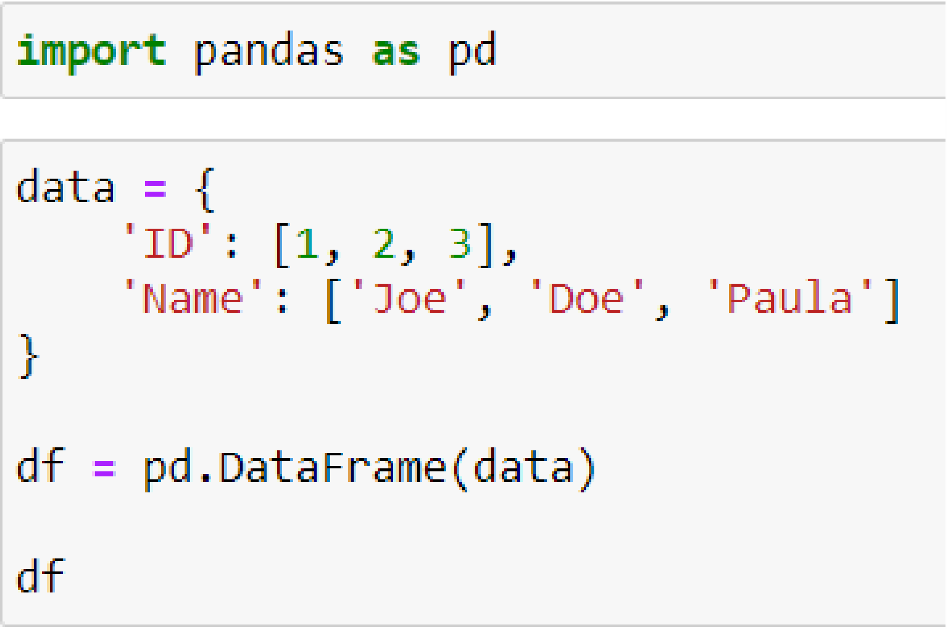
LESS THAN (<)



```sql
CREATE TABLE "df"
(
"ID" INTEGER,
"NAME" VARCHAR(10)
);

INSERT INTO "df" VALUES
(1, 'Joe'),
(2, 'Doe'),
(3, 'Paula');

SELECT *
FROM "df"
WHERE "ID" < '2';
```

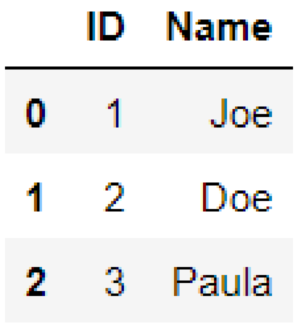


...	ID	NAME
	1	Joe

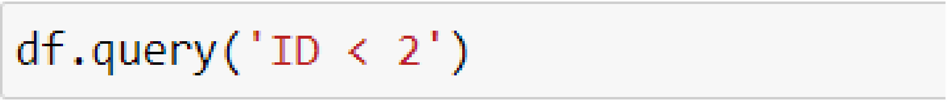


```python
import pandas as pd
```

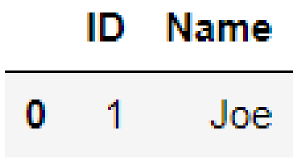
```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Doe', 'Paula']
}

df = pd.DataFrame(data)

df
```



	ID	Name
0	1	Joe
1	2	Doe
2	3	Paula



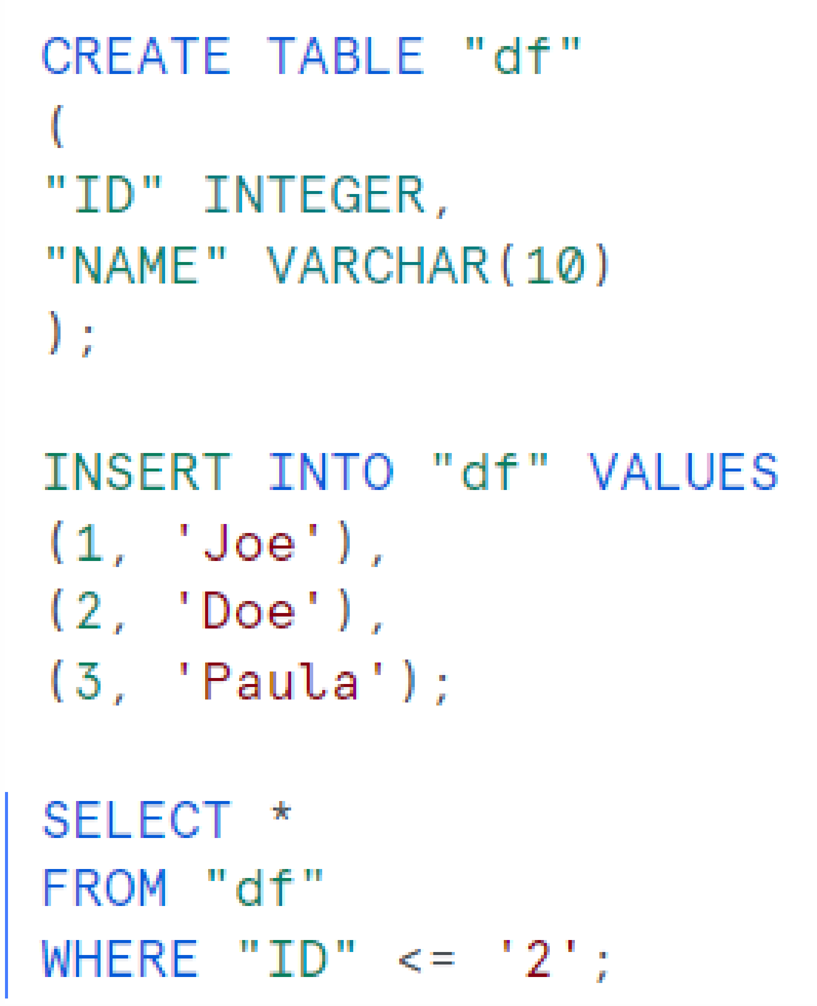
```python
df.query('ID < 2')
```



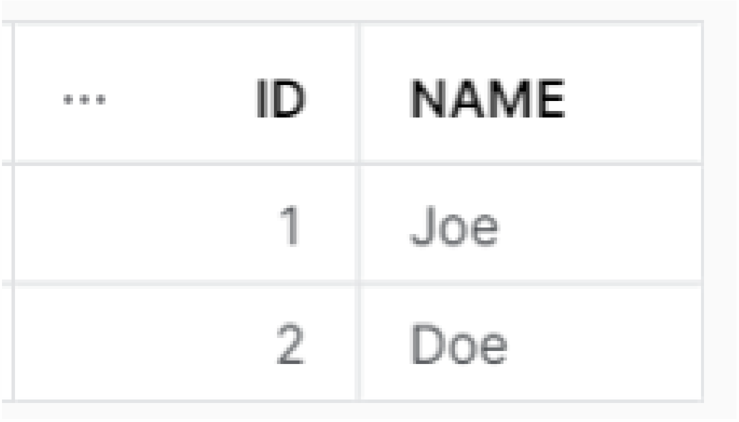
	ID	Name
0	1	Joe

Uzwal Goud Vaddeboina

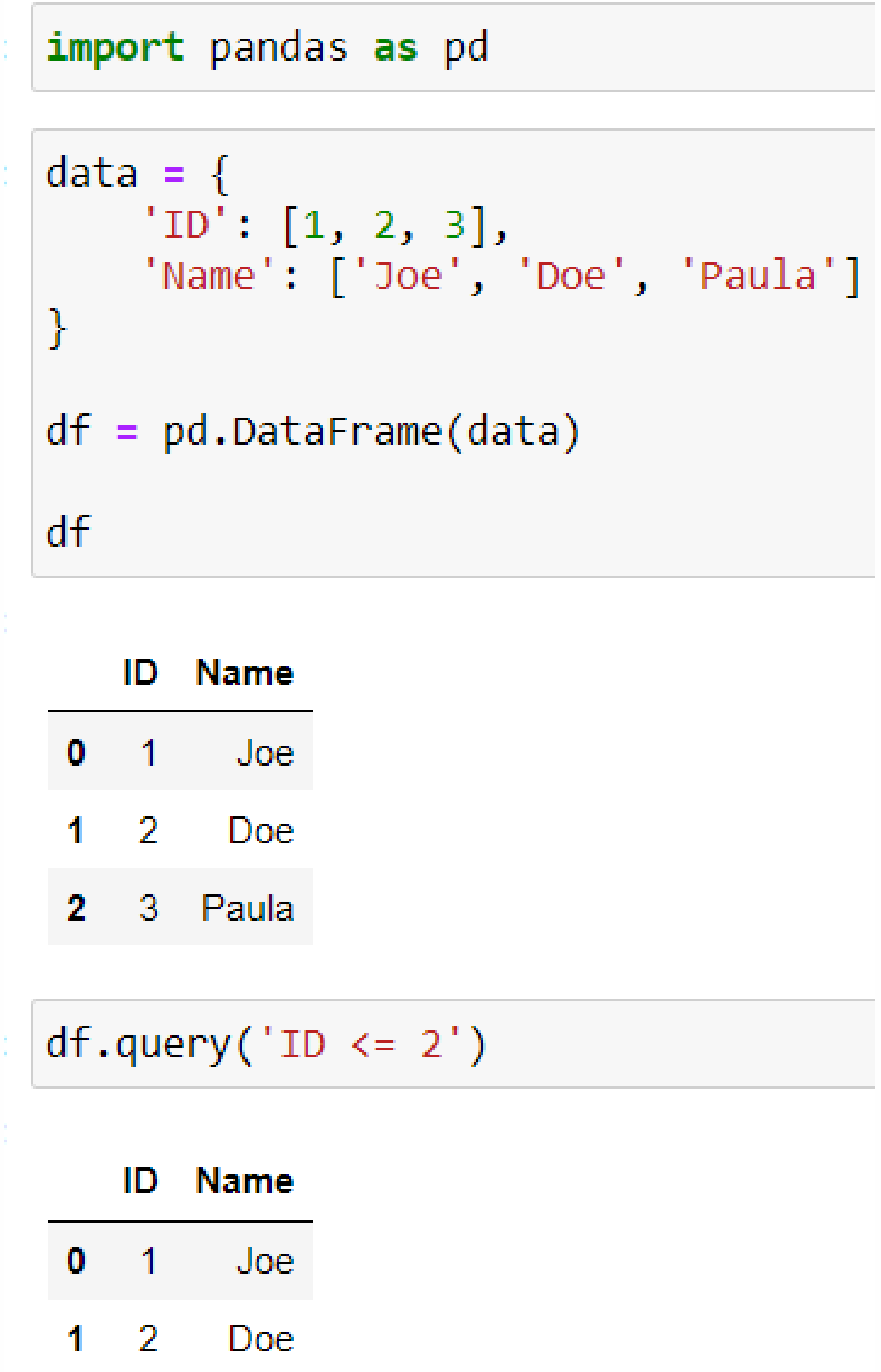
LESS THAN EQUAL TO (<=)



```sql
CREATE TABLE "df"
(
"ID" INTEGER,
"NAME" VARCHAR(10)
);

INSERT INTO "df" VALUES
(1, 'Joe'),
(2, 'Doe'),
(3, 'Paula');

SELECT *
FROM "df"
WHERE "ID" <= '2';
```



...	ID	NAME
	1	Joe
	2	Doe



```python
import pandas as pd
```

```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Doe', 'Paula']
}

df = pd.DataFrame(data)

df
```

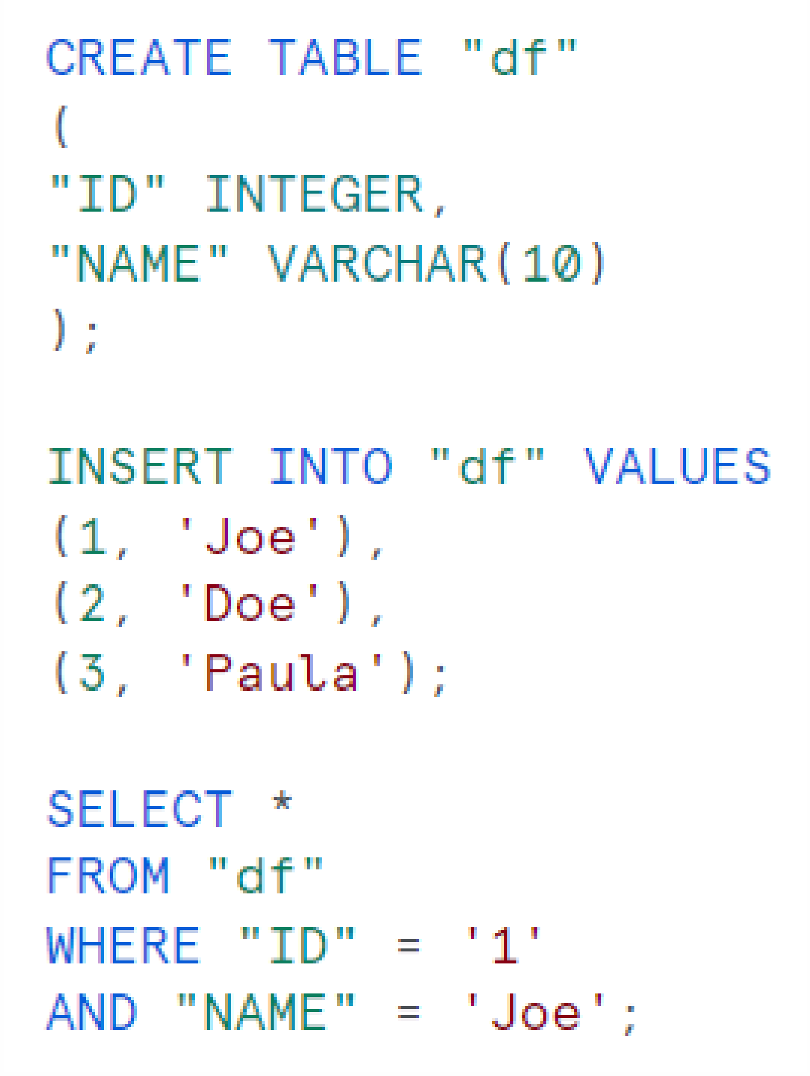
	ID	Name
0	1	Joe
1	2	Doe
2	3	Paula

```python
df.query('ID <= 2')
```

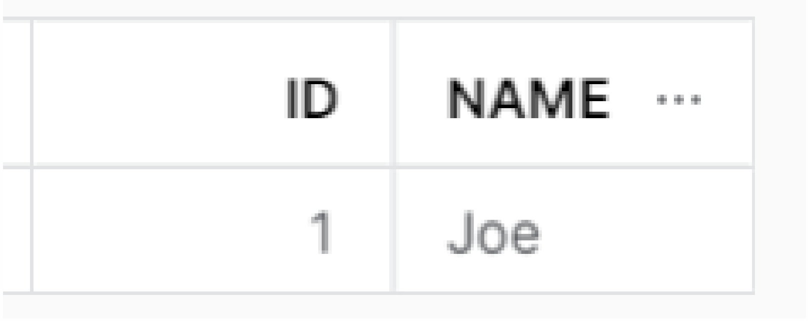
	ID	Name
0	1	Joe
1	2	Doe

Uzwal Goud Vaddeboina

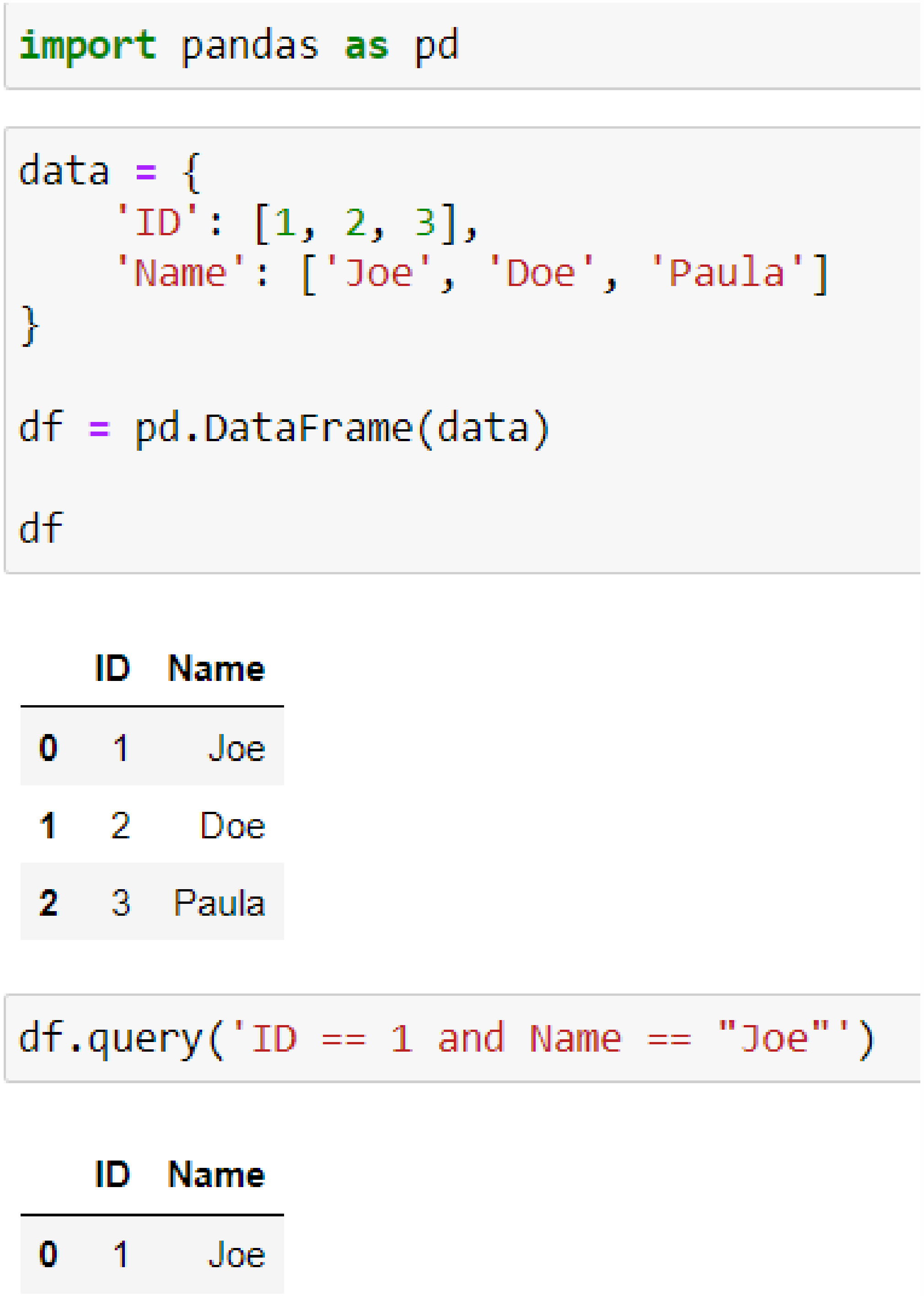
AND



```sql
CREATE TABLE "df"
(
"ID" INTEGER,
"NAME" VARCHAR(10)
);

INSERT INTO "df" VALUES
(1, 'Joe'),
(2, 'Doe'),
(3, 'Paula');

SELECT *
FROM "df"
WHERE "ID" = '1'
AND "NAME" = 'Joe';
```



ID	NAME	···
1	Joe	



```python
import pandas as pd
```

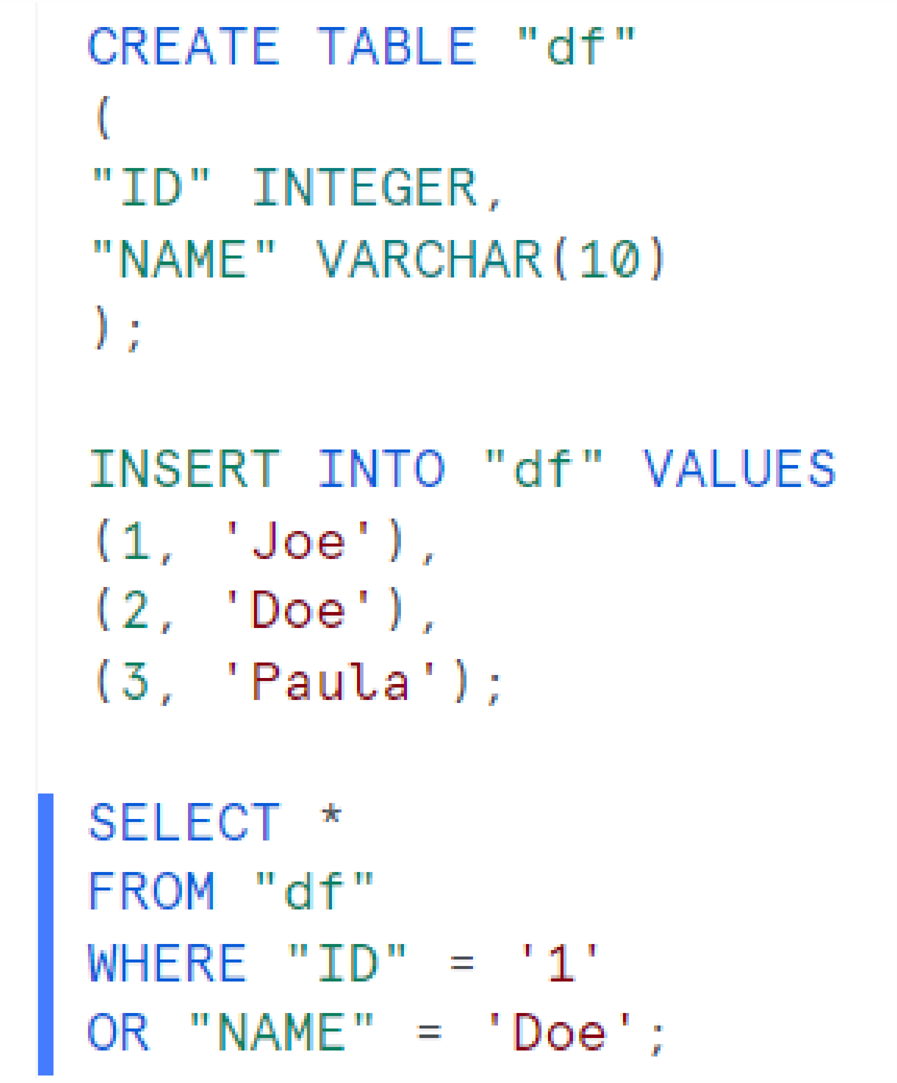
```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Doe', 'Paula']
}

df = pd.DataFrame(data)

df
```

	ID	Name
0	1	Joe
1	2	Doe
2	3	Paula

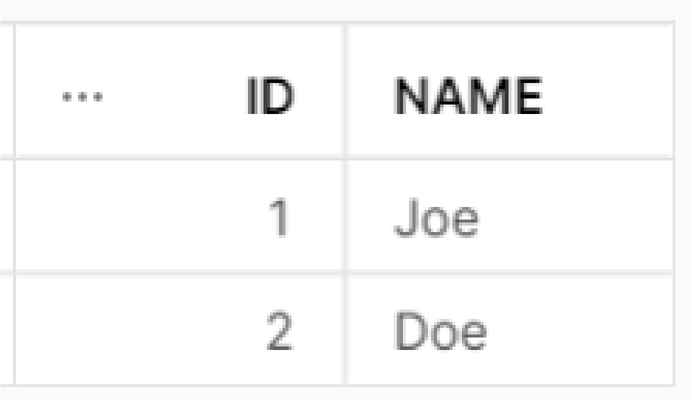
```python
df.query('ID == 1 and Name == "Joe"')
```

	ID	Name
0	1	Joe

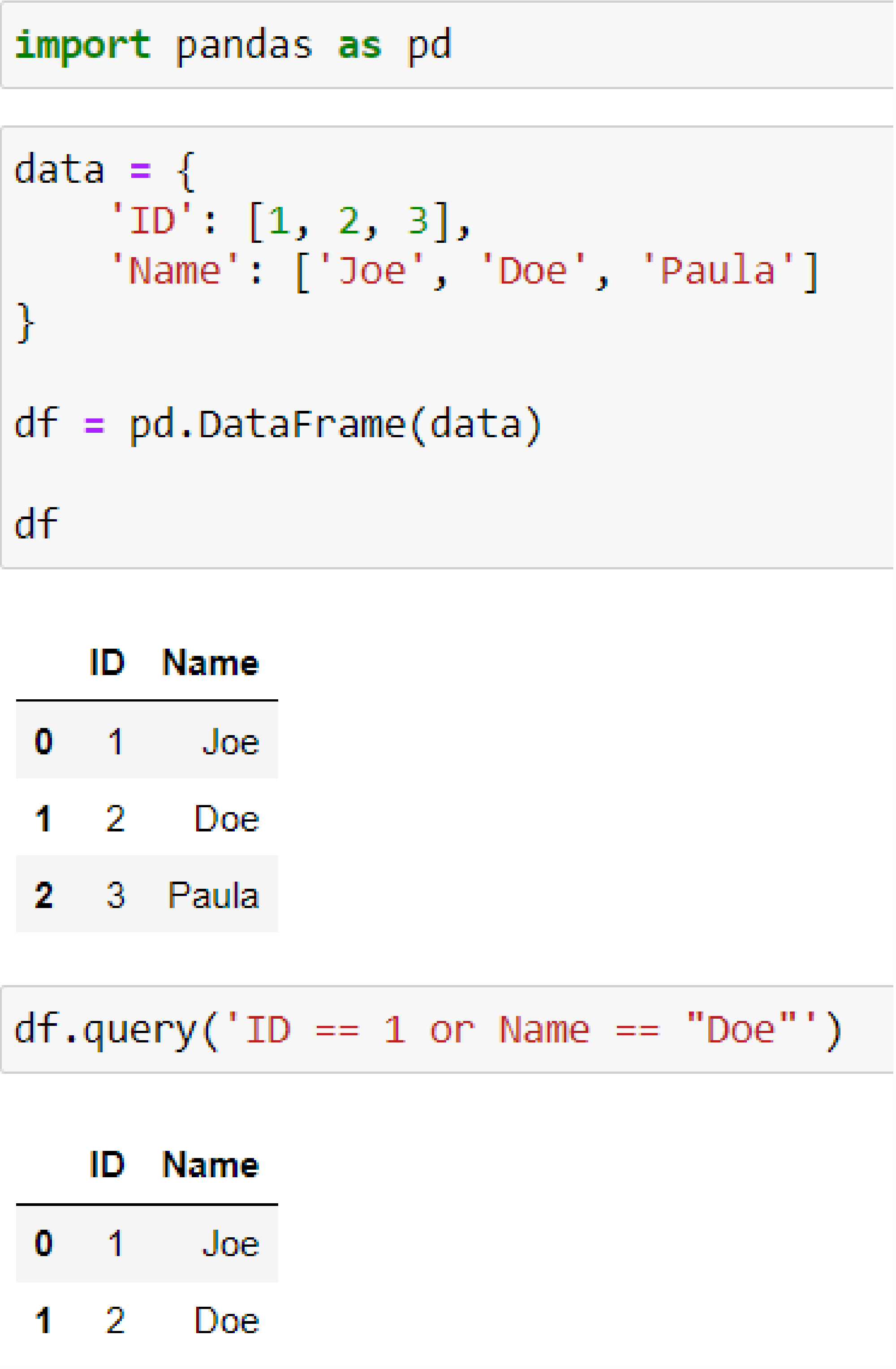
OR



```sql
CREATE TABLE "df"
(
"ID" INTEGER,
"NAME" VARCHAR(10)
);

INSERT INTO "df" VALUES
(1, 'Joe'),
(2, 'Doe'),
(3, 'Paula');

SELECT *
FROM "df"
WHERE "ID" = '1'
OR "NAME" = 'Doe';
```



...	ID	NAME
	1	Joe
	2	Doe



```python
import pandas as pd
```

```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Doe', 'Paula']
}

df = pd.DataFrame(data)

df
```

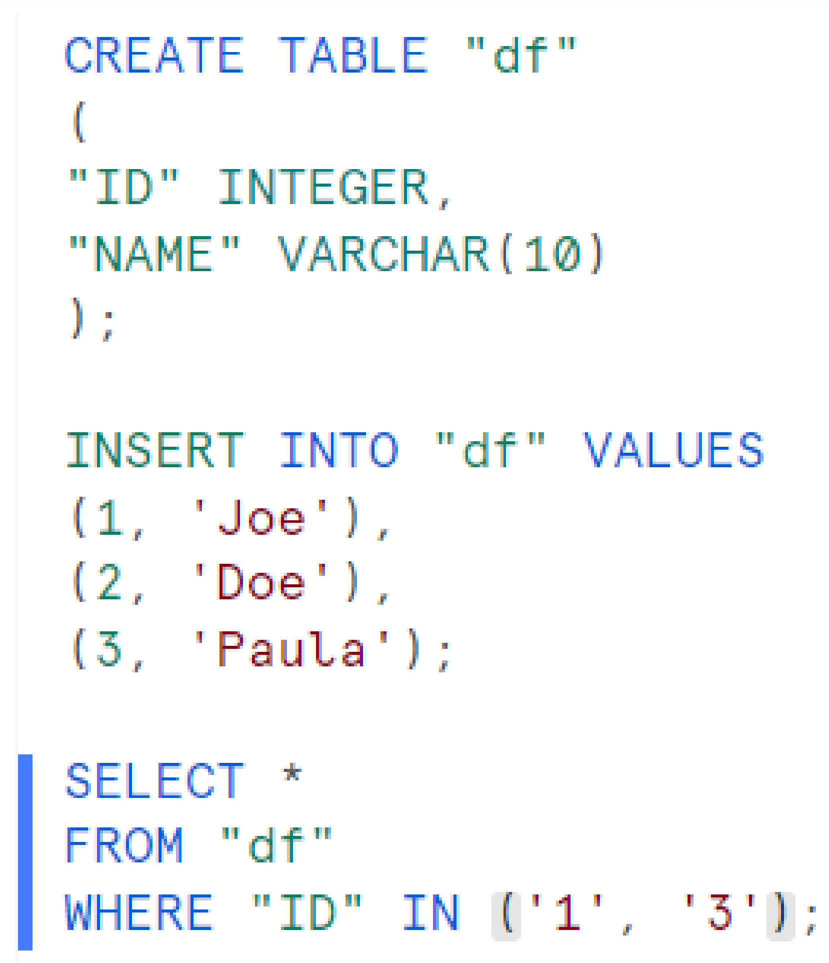
	ID	Name
0	1	Joe
1	2	Doe
2	3	Paula

```python
df.query('ID == 1 or Name == "Doe"')
```

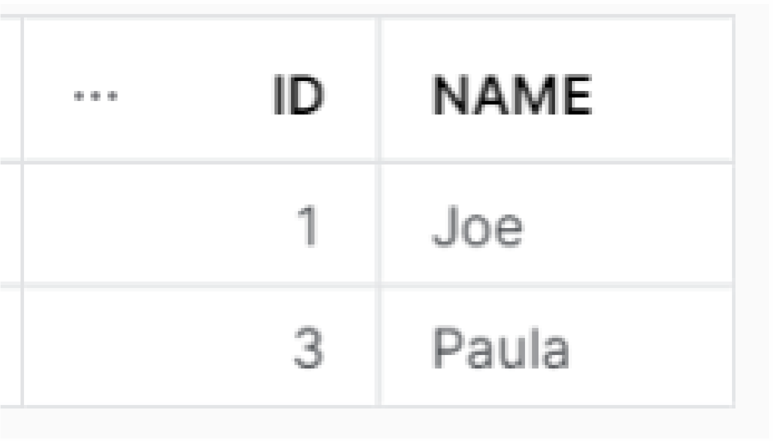
	ID	Name
0	1	Joe
1	2	Doe

in Uzwal Goud Vaddeboina

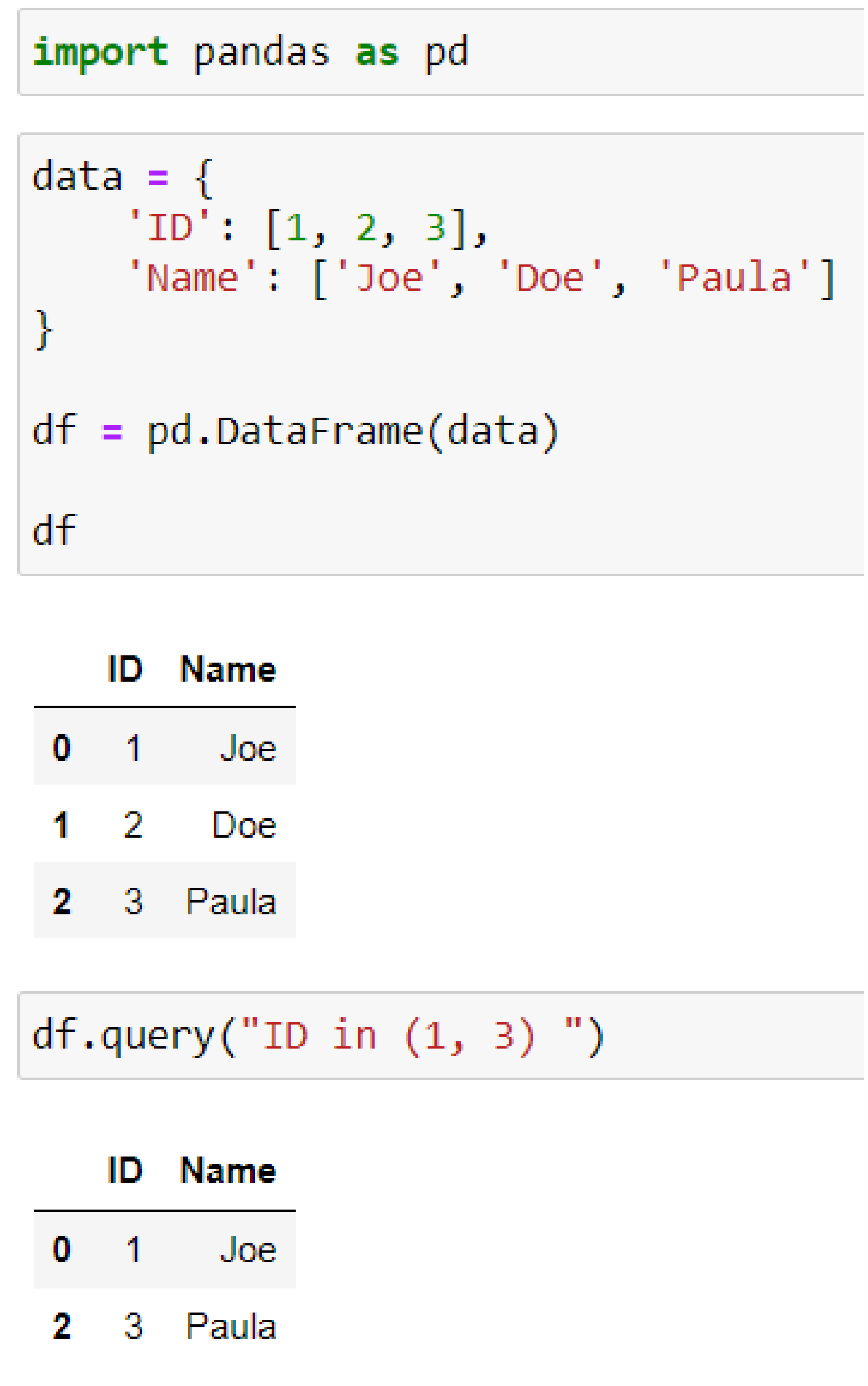
IN



```sql
CREATE TABLE "df"
(
"ID" INTEGER,
"NAME" VARCHAR(10)
);

INSERT INTO "df" VALUES
(1, 'Joe'),
(2, 'Doe'),
(3, 'Paula');

SELECT *
FROM "df"
WHERE "ID" IN ('1', '3');
```



...	ID	NAME
	1	Joe
	3	Paula



```python
import pandas as pd
```

```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Doe', 'Paula']
}

df = pd.DataFrame(data)

df
```

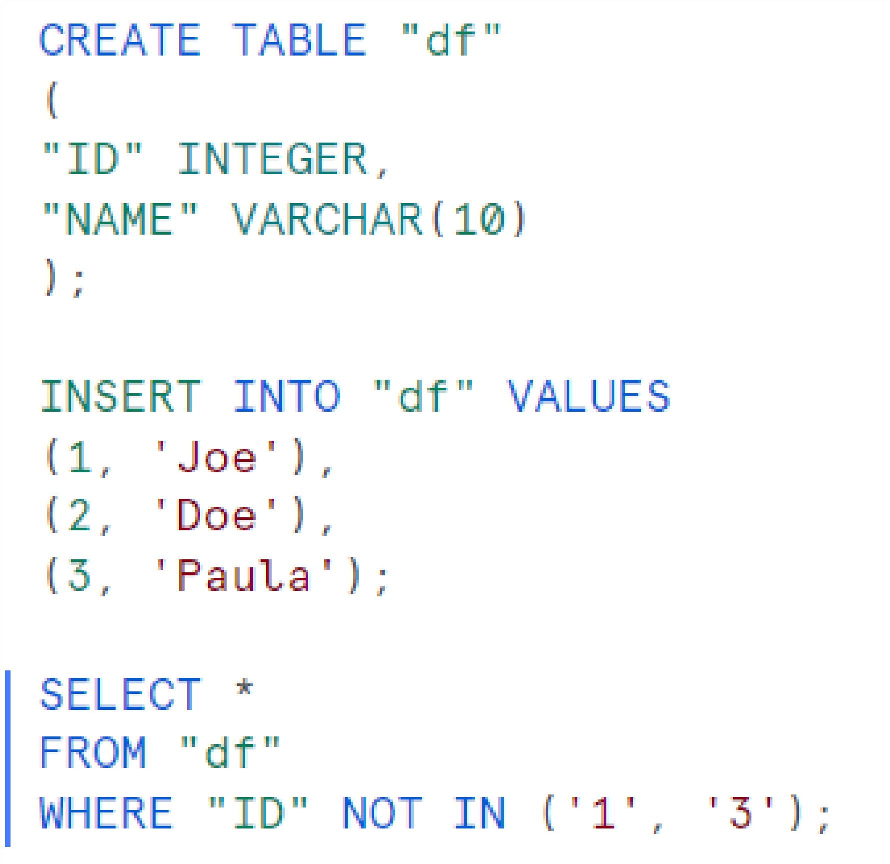
	ID	Name
0	1	Joe
1	2	Doe
2	3	Paula

```python
df.query("ID in (1, 3) ")
```

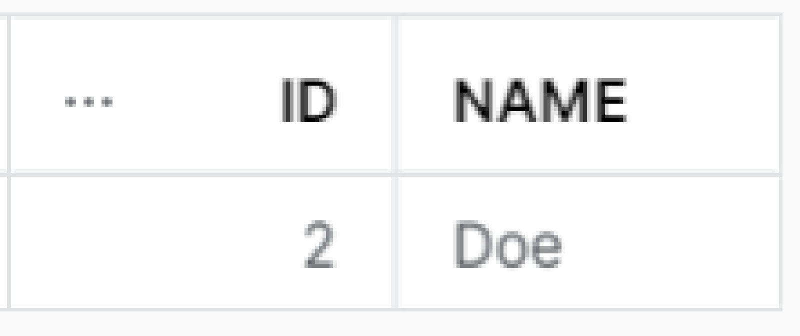
	ID	Name
0	1	Joe
2	3	Paula

Uzwal Goud Vaddeboina

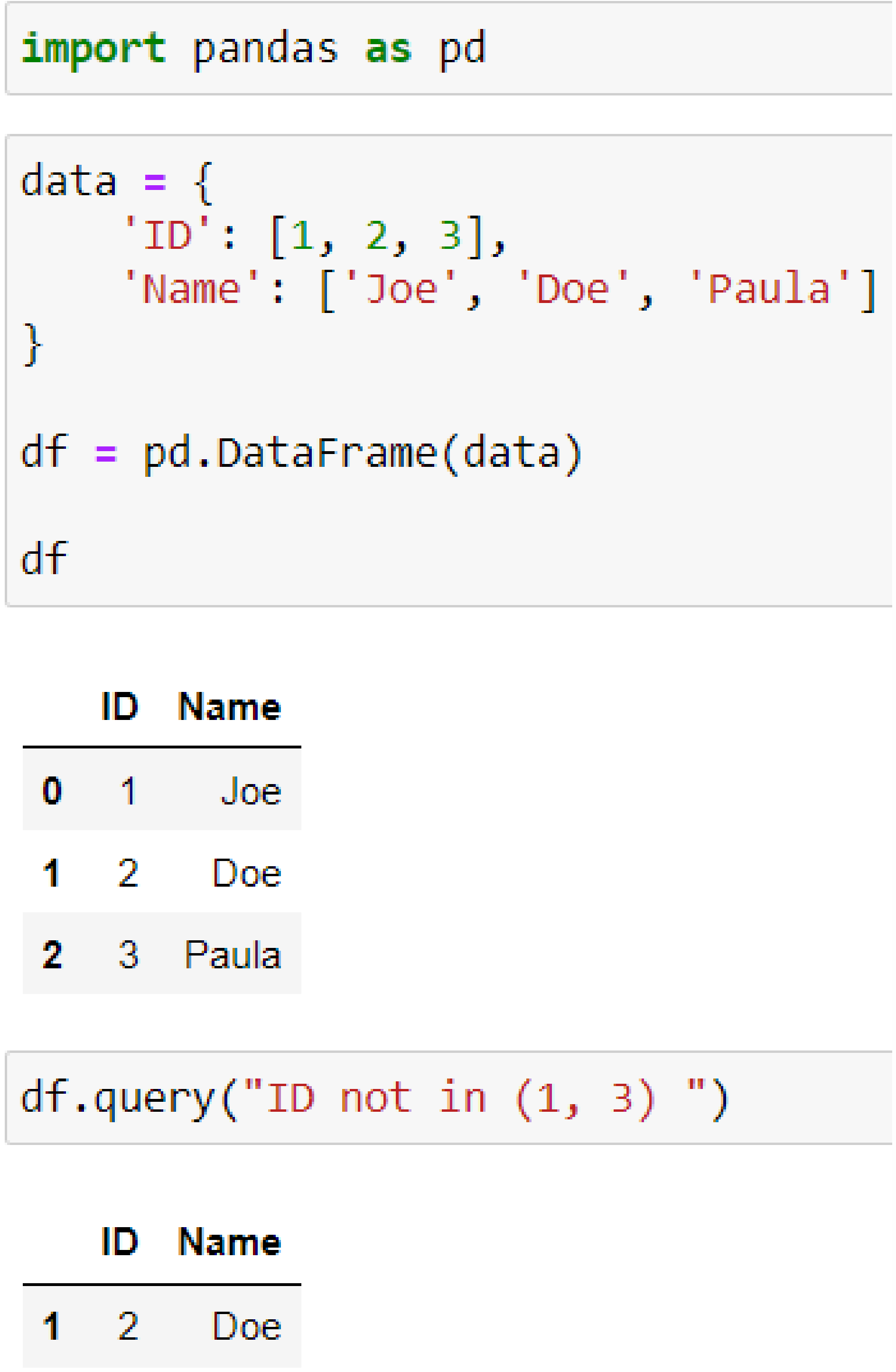
NOT IN



```sql
CREATE TABLE "df"
(
"ID" INTEGER,
"NAME" VARCHAR(10)
);

INSERT INTO "df" VALUES
(1, 'Joe'),
(2, 'Doe'),
(3, 'Paula');

SELECT *
FROM "df"
WHERE "ID" NOT IN ('1', '3');
```



... ID	NAME
2	Doe



```python
import pandas as pd
```

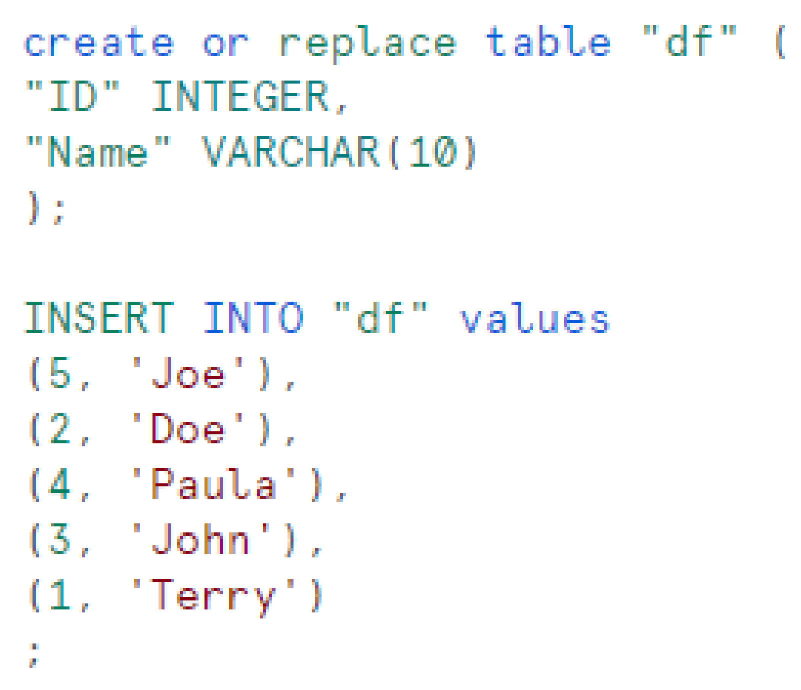
```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Doe', 'Paula']
}

df = pd.DataFrame(data)

df
```

	ID	Name
0	1	Joe
1	2	Doe
2	3	Paula

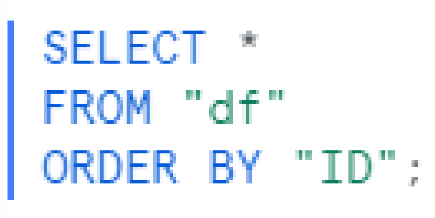
```python
df.query("ID not in (1, 3) ")
```

	ID	Name
1	2	Doe

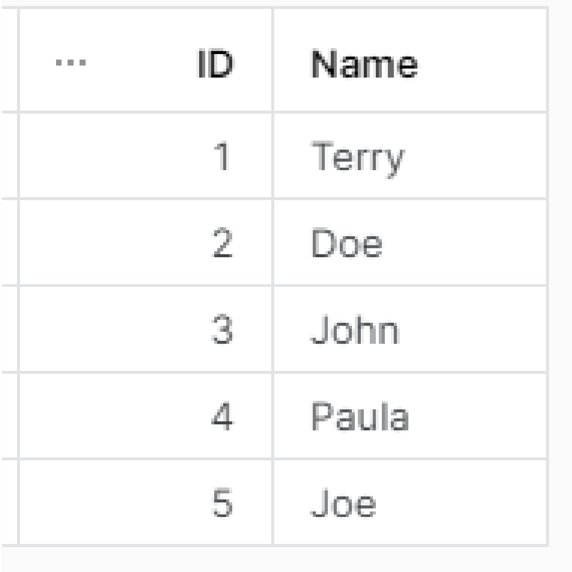
SORT BY SINGLE COLUMN ASC



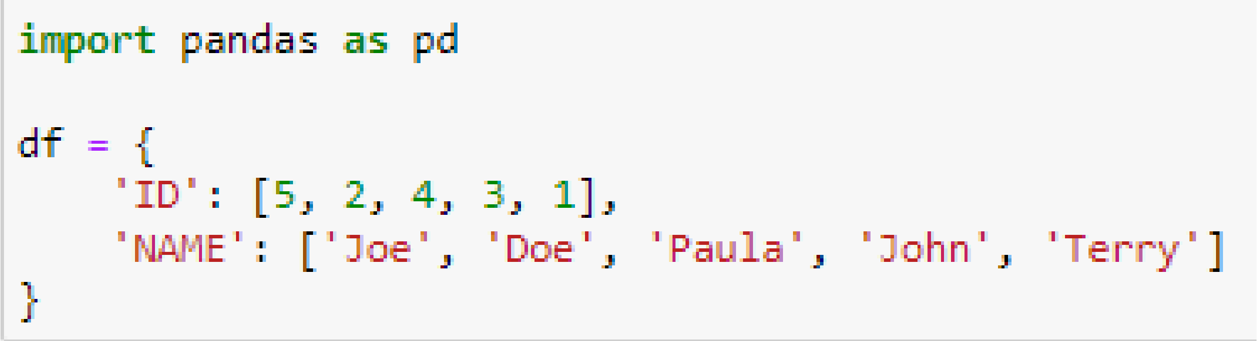
```sql
create or replace table "df" (
"ID" INTEGER,
"Name" VARCHAR(10)
);

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



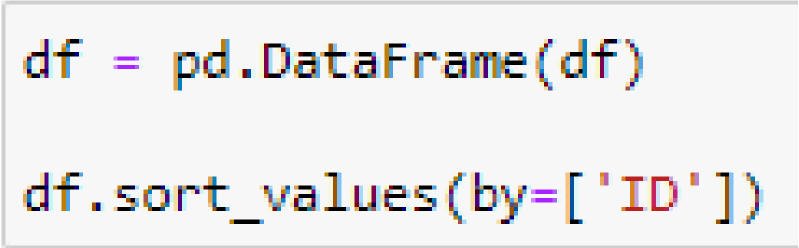
```sql
SELECT *
FROM "df"
ORDER BY "ID";
```



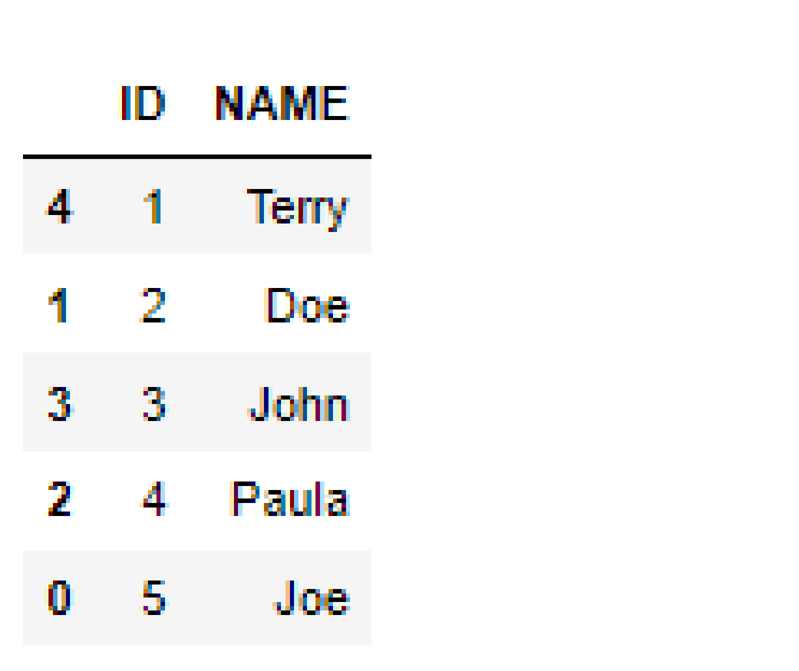
...	ID	Name
	1	Terry
	2	Doe
	3	John
	4	Paula
	5	Joe



```python
import pandas as pd

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



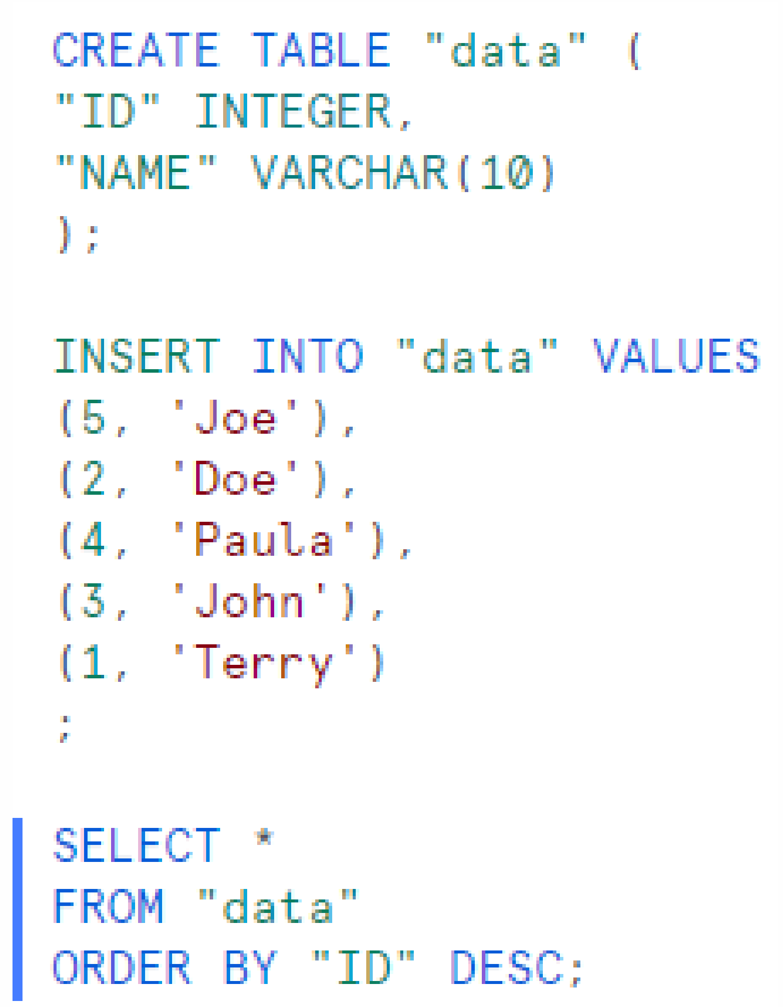
```python
df = pd.DataFrame(df)

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



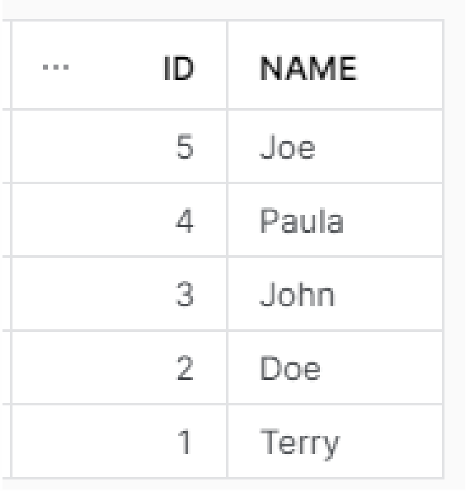
	ID	NAME
4	1	Terry
1	2	Doe
3	3	John
2	4	Paula
0	5	Joe

in Uzwal Goud Vaddeboina

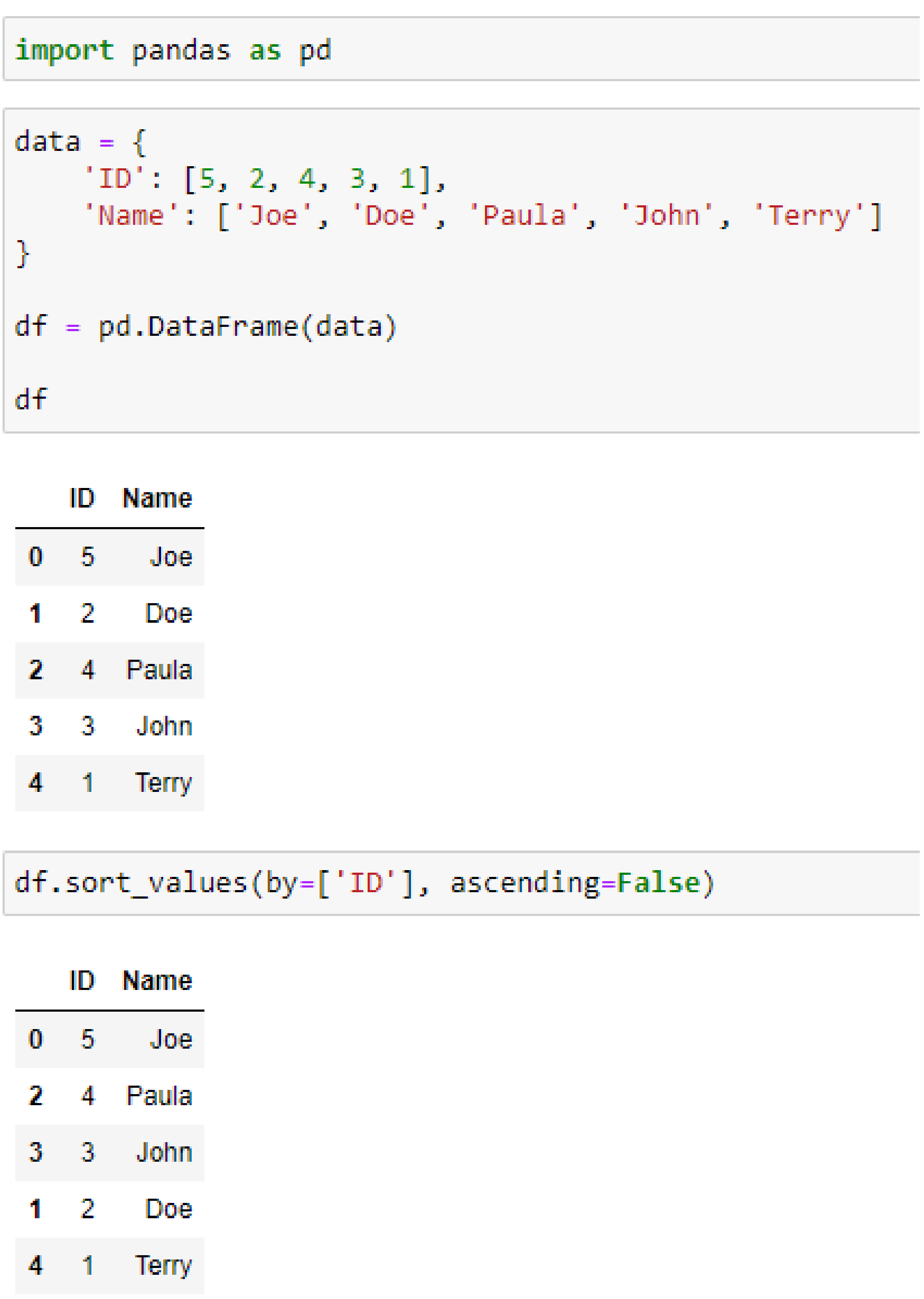
SORT BY SINGLE COLUMN DESC



```sql
CREATE TABLE "data" (
"ID" INTEGER,
"NAME" VARCHAR(10)
);

INSERT INTO "data" VALUES
(5, 'Joe'),
(2, 'Doe'),
(4, 'Paula'),
(3, 'John'),
(1, 'Terry')
;

SELECT *
FROM "data"
ORDER BY "ID" DESC;
```



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



```python
import pandas as pd
```

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

df = pd.DataFrame(data)

df
```

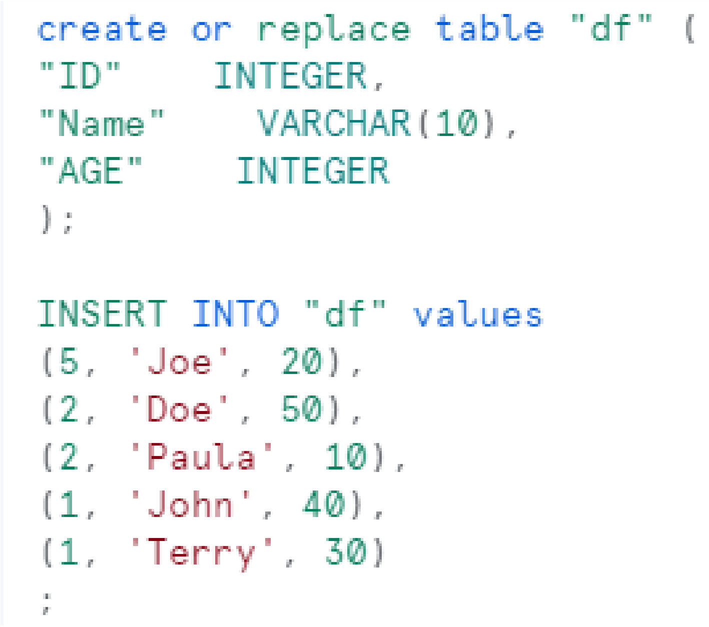
	ID	Name
0	5	Joe
1	2	Doe
2	4	Paula
3	3	John
4	1	Terry

```python
df.sort_values(by=['ID'], ascending=False)
```

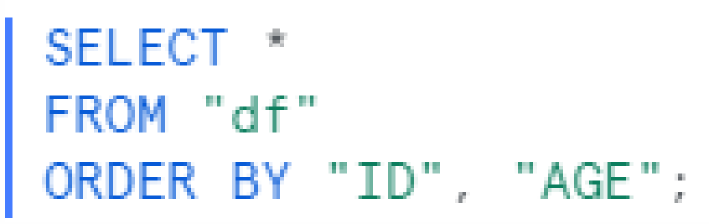
	ID	Name
0	5	Joe
2	4	Paula
3	3	John
1	2	Doe
4	1	Terry

Uzwal Goud Vaddeboina

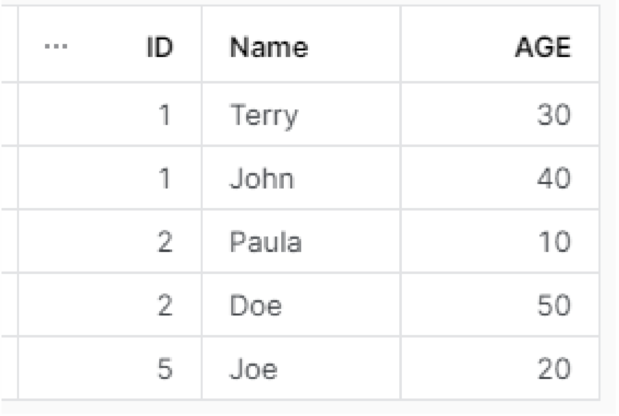
SORT BY MULTIPLE COLUMNS ASC



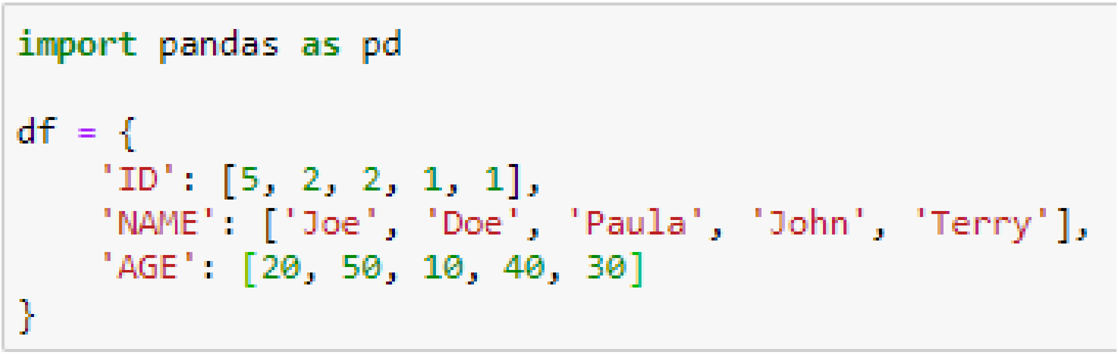
```
create or replace table "df" (
"ID"     INTEGER,
"Name"     VARCHAR(10),
"AGE"     INTEGER
);

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



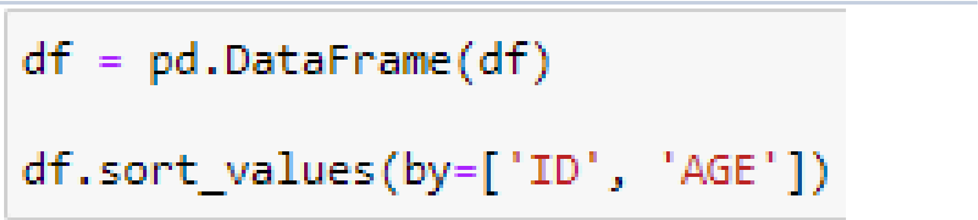
```
SELECT *
FROM "df"
ORDER BY "ID", "AGE";
```



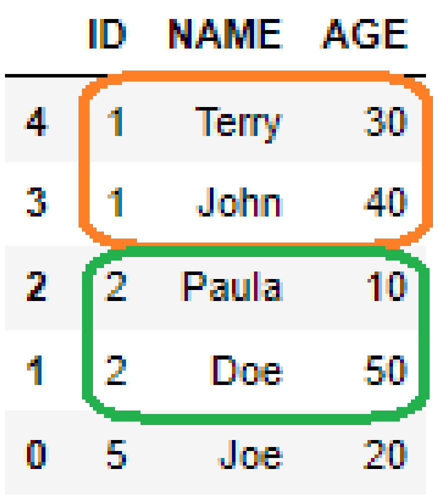
...	ID	Name	AGE
	1	Terry	30
	1	John	40
	2	Paula	10
	2	Doe	50
	5	Joe	20



```
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'])
```

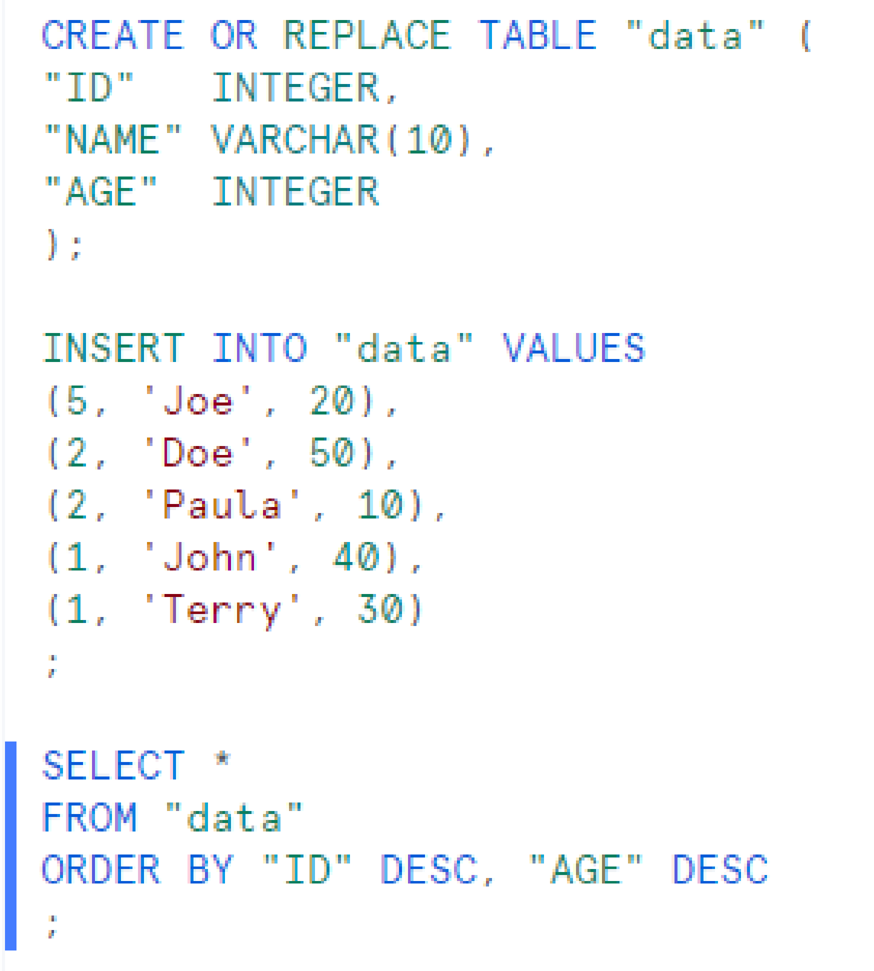


	ID	NAME	AGE
4	1	Terry	30
3	1	John	40
2	2	Paula	10
1	2	Doe	50
0	5	Joe	20

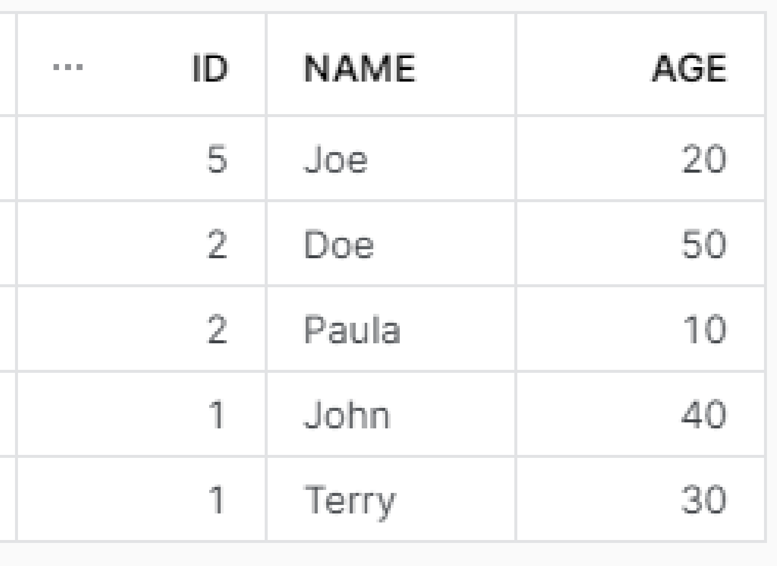
Sorted by default in ascending order

Uzwal Goud Vaddeboina

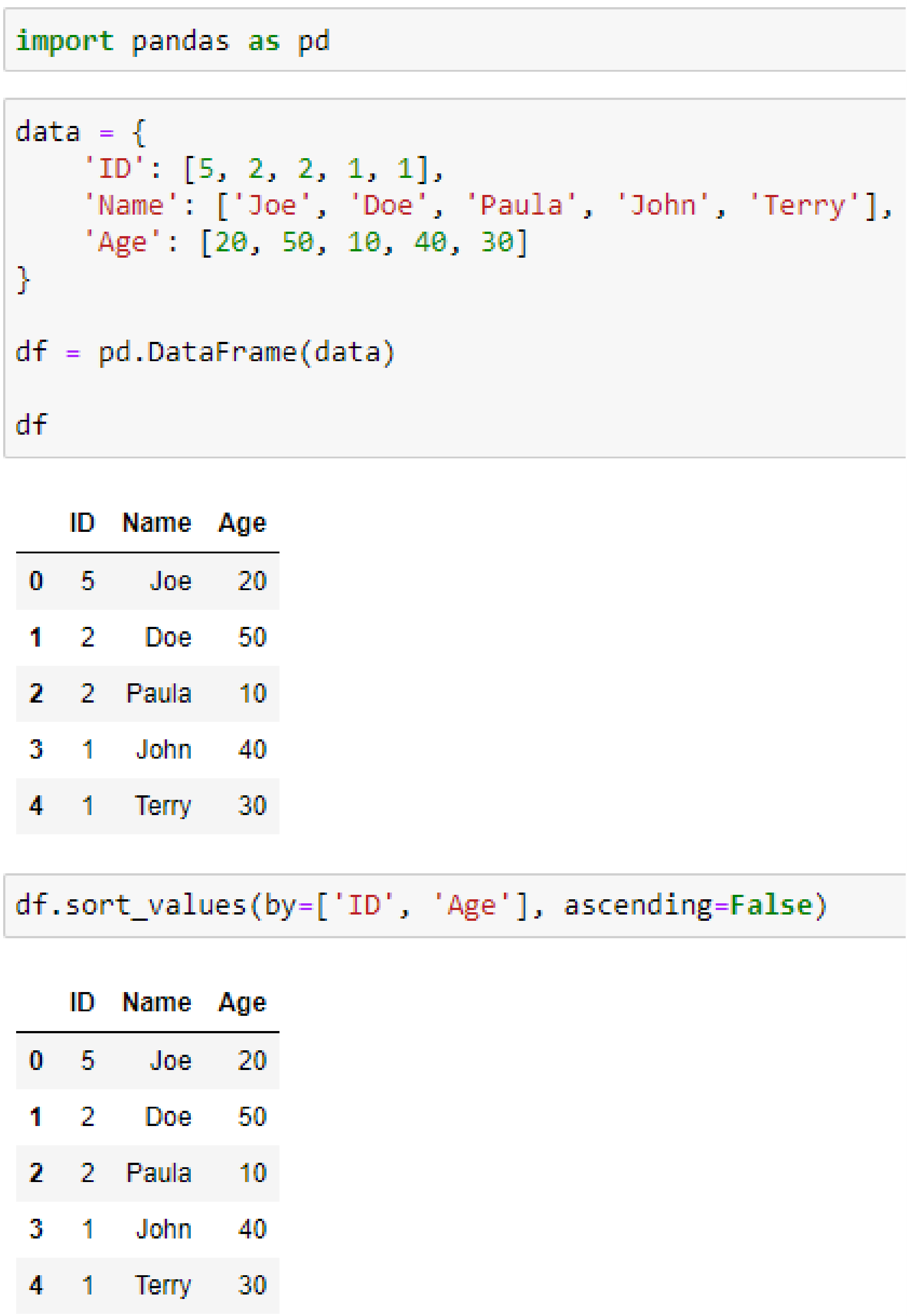
SORT BY MULTIPLE COLUMNS DESC



```sql
CREATE OR REPLACE TABLE "data" (
"ID"   INTEGER,
"NAME" VARCHAR(10),
"AGE"  INTEGER
);

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

SELECT *
FROM "data"
ORDER BY "ID" DESC, "AGE" DESC
;
```



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



```python
import pandas as pd
```

```python
data = {
    'ID': [5, 2, 2, 1, 1],
    'Name': ['Joe', 'Doe', 'Paula', 'John', 'Terry'],
    'Age': [20, 50, 10, 40, 30]
}

df = pd.DataFrame(data)

df
```

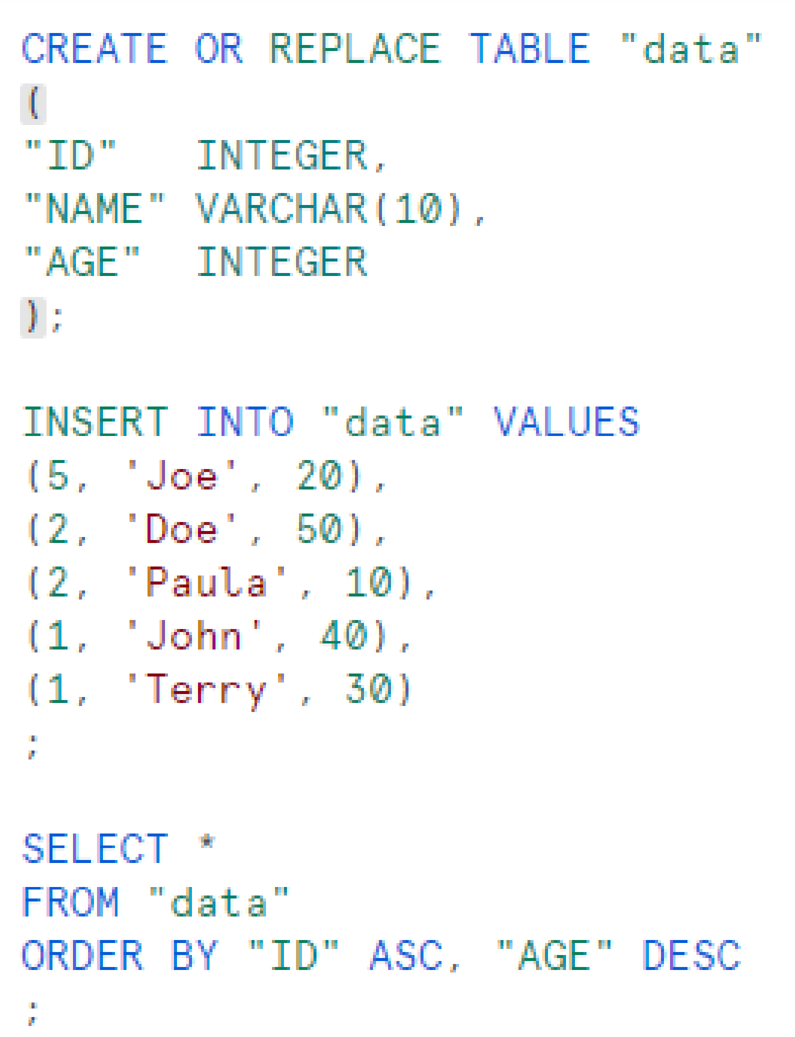
```
   ID   Name  Age
0   5    Joe   20
1   2    Doe   50
2   2  Paula   10
3   1   John   40
4   1  Terry   30
```

```python
df.sort_values(by=['ID', 'Age'], ascending=False)
```

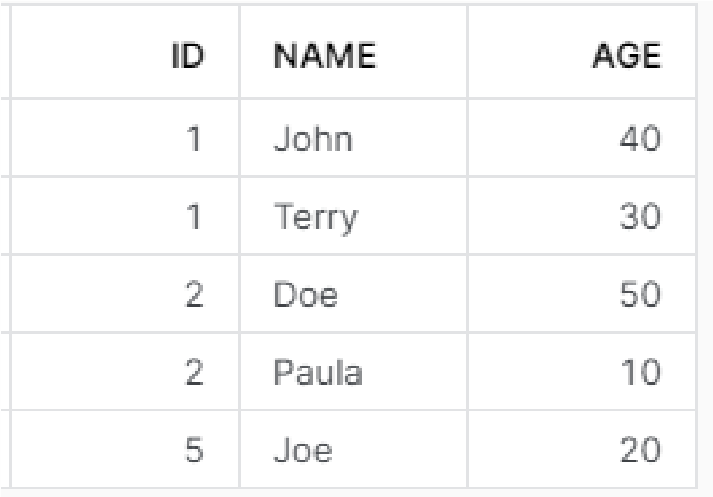
```
   ID   Name  Age
0   5    Joe   20
1   2    Doe   50
2   2  Paula   10
3   1   John   40
4   1  Terry   30
```

Uzwal Goud Vaddeboina

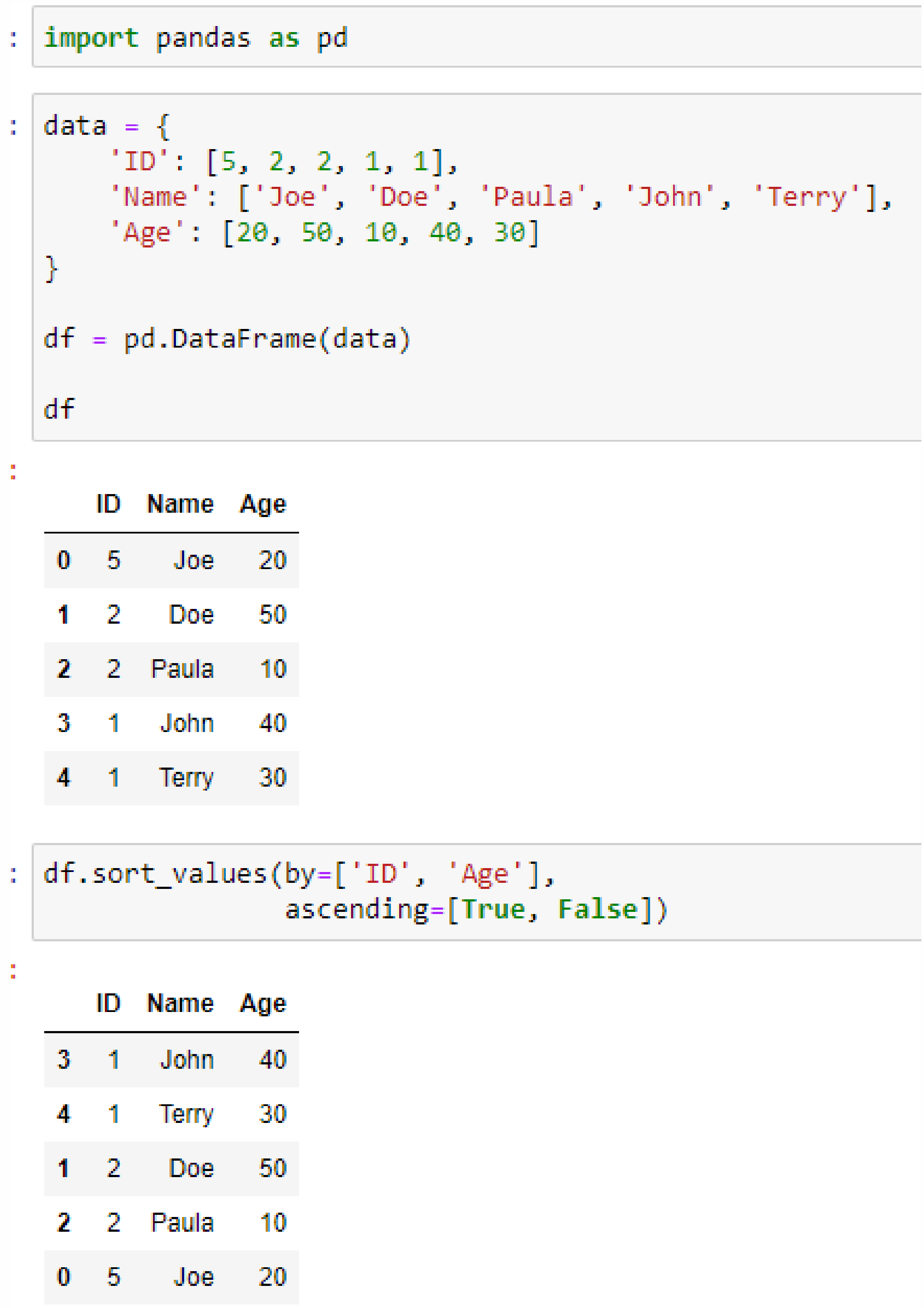
SORT BY ASC AND DESC



```sql
CREATE OR REPLACE TABLE "data"
(
"ID"    INTEGER,
"NAME"  VARCHAR(10),
"AGE"   INTEGER
);

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

SELECT *
FROM "data"
ORDER BY "ID" ASC, "AGE" DESC
;
```



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



```python
import pandas as pd
```

```python
data = {
    'ID': [5, 2, 2, 1, 1],
    'Name': ['Joe', 'Doe', 'Paula', 'John', 'Terry'],
    'Age': [20, 50, 10, 40, 30]
}

df = pd.DataFrame(data)

df
```

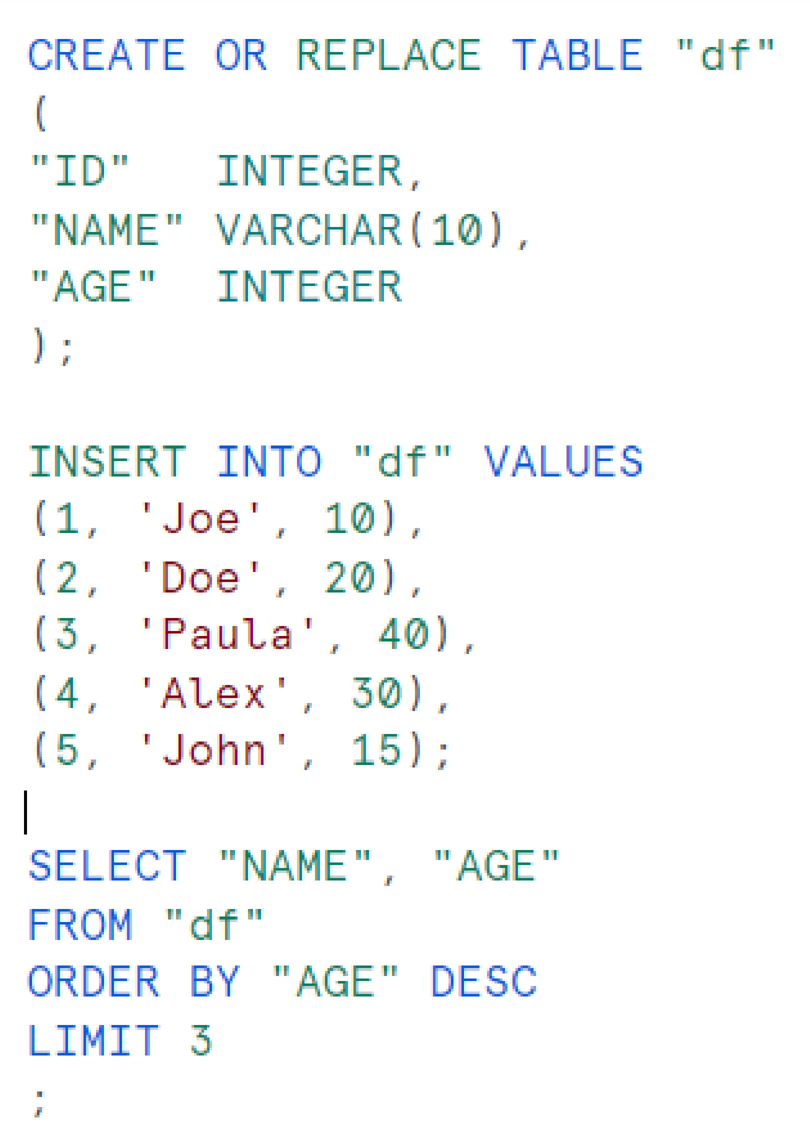
	ID	Name	Age
0	5	Joe	20
1	2	Doe	50
2	2	Paula	10
3	1	John	40
4	1	Terry	30

```python
df.sort_values(by=['ID', 'Age'],
               ascending=[True, False])
```

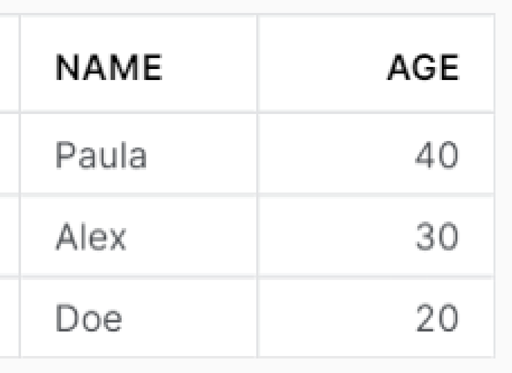
	ID	Name	Age
3	1	John	40
4	1	Terry	30
1	2	Doe	50
2	2	Paula	10
0	5	Joe	20

Uzwal Goud Vaddeboina

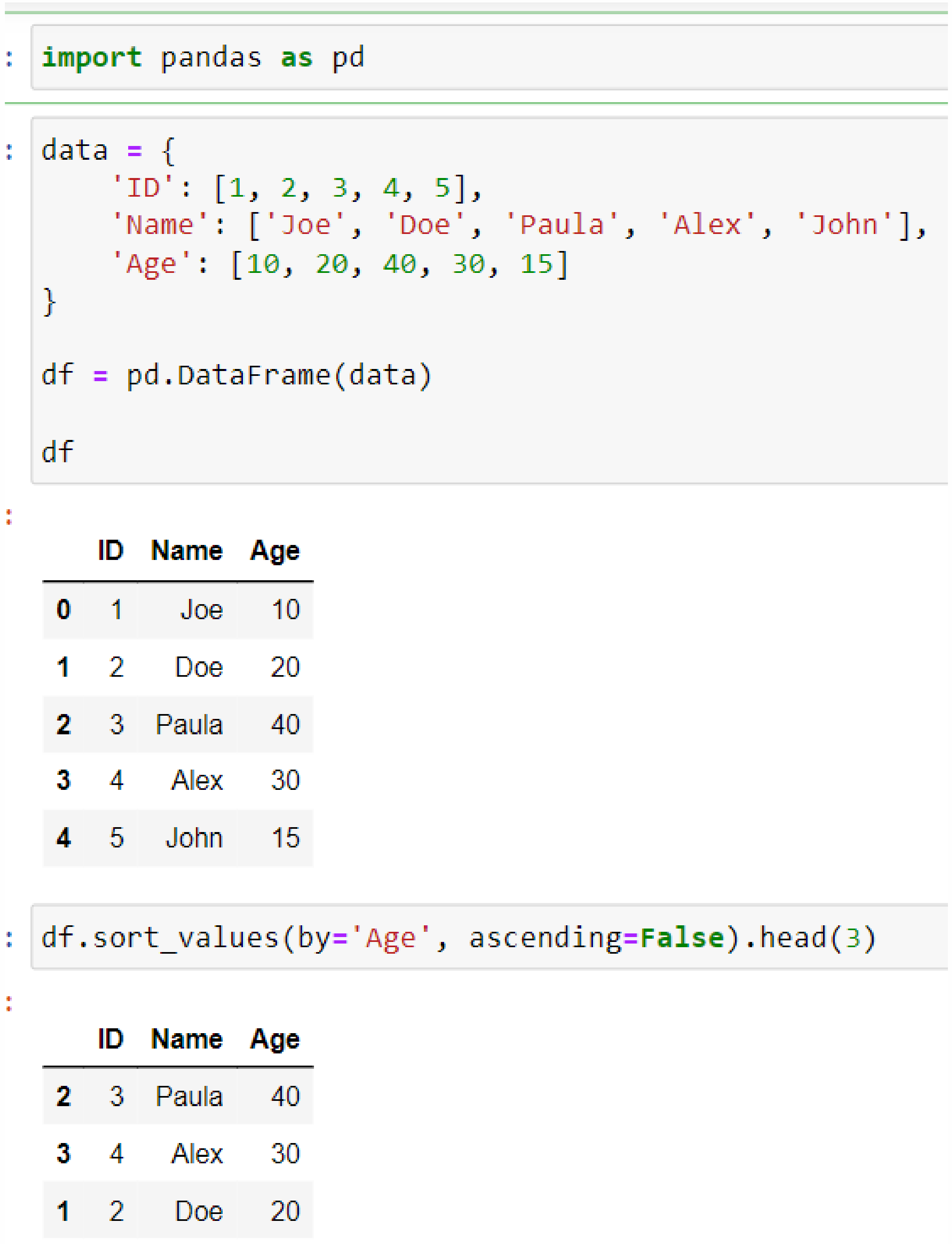
TOP N ROWS



```sql
CREATE OR REPLACE TABLE "df"
(
"ID"   INTEGER,
"NAME" VARCHAR(10),
"AGE"  INTEGER
);

INSERT INTO "df" VALUES
(1, 'Joe', 10),
(2, 'Doe', 20),
(3, 'Paula', 40),
(4, 'Alex', 30),
(5, 'John', 15);

SELECT "NAME", "AGE"
FROM "df"
ORDER BY "AGE" DESC
LIMIT 3
;
```



NAME	AGE
Paula	40
Alex	30
Doe	20



```python
import pandas as pd
```

```python
data = {
    'ID': [1, 2, 3, 4, 5],
    'Name': ['Joe', 'Doe', 'Paula', 'Alex', 'John'],
    'Age': [10, 20, 40, 30, 15]
}

df = pd.DataFrame(data)

df
```

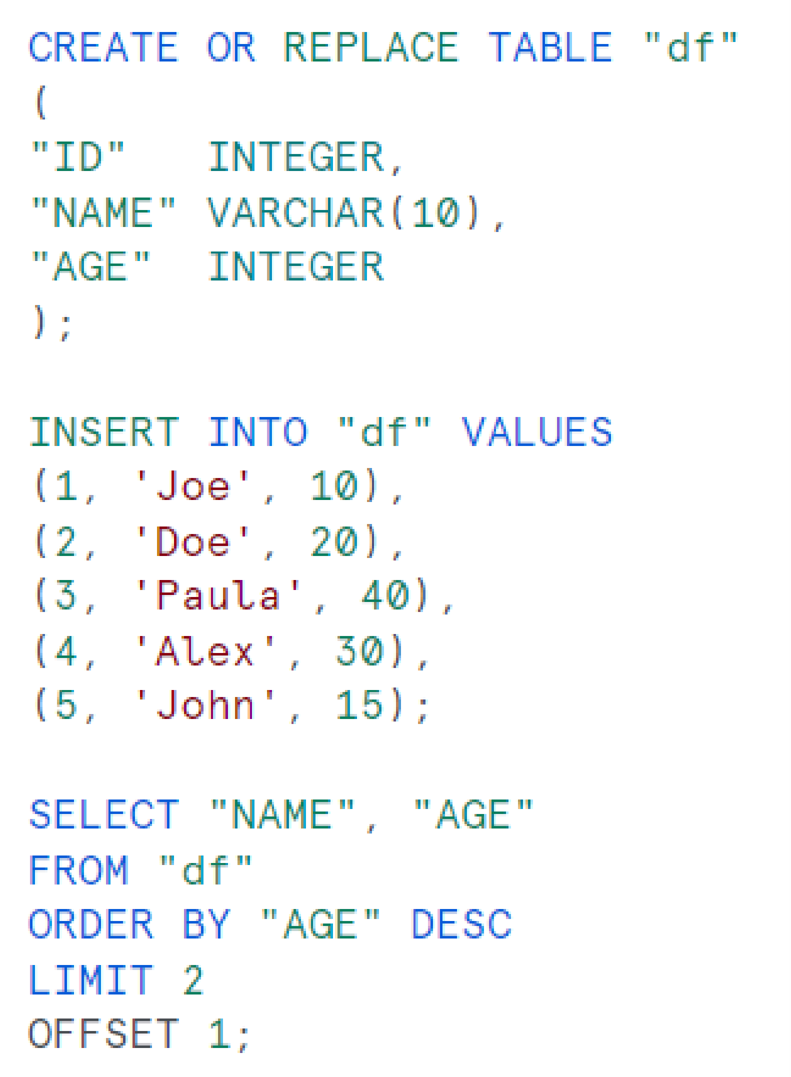
```
   ID  Name   Age
0  1   Joe    10
1  2   Doe    20
2  3   Paula  40
3  4   Alex   30
4  5   John   15
```

```python
df.sort_values(by='Age', ascending=False).head(3)
```

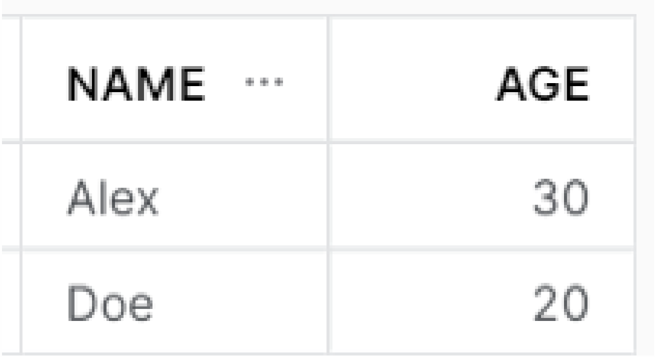
```
   ID  Name   Age
2  3   Paula  40
3  4   Alex   30
1  2   Doe    20
```

Uzwal Goud Vaddeboina

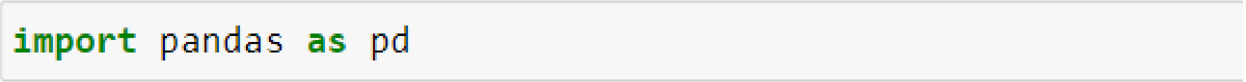
OFFSET



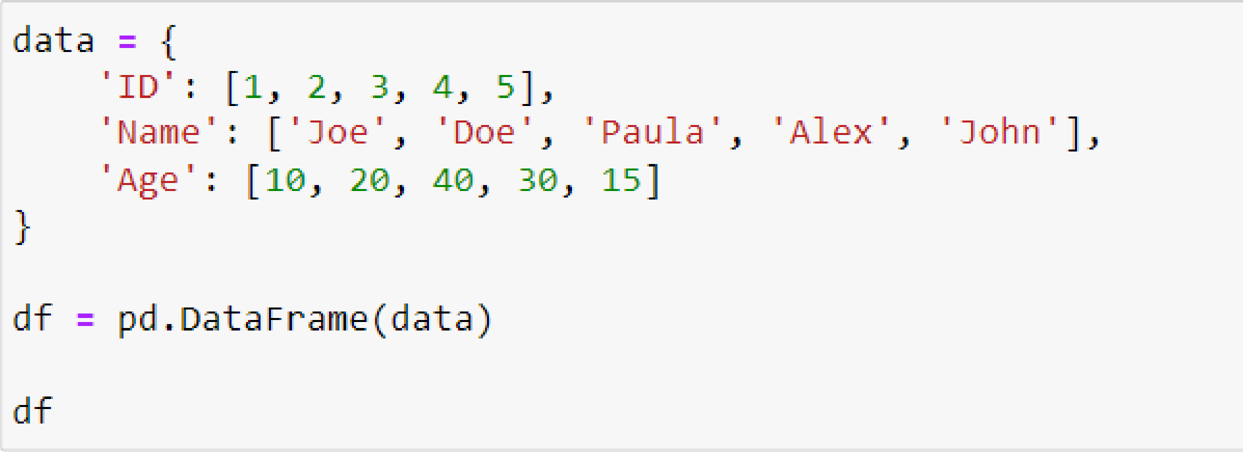
```sql
CREATE OR REPLACE TABLE "df"
(
"ID"    INTEGER,
"NAME"  VARCHAR(10),
"AGE"   INTEGER
);

INSERT INTO "df" VALUES
(1, 'Joe', 10),
(2, 'Doe', 20),
(3, 'Paula', 40),
(4, 'Alex', 30),
(5, 'John', 15);

SELECT "NAME", "AGE"
FROM "df"
ORDER BY "AGE" DESC
LIMIT 2
OFFSET 1;
```



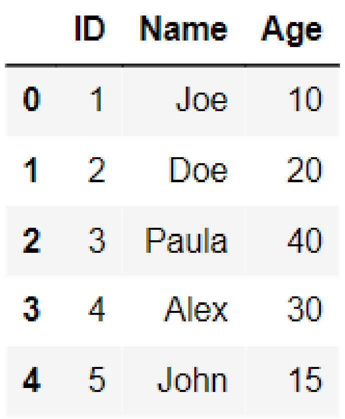
NAME ···	AGE
Alex	30
Doe	20



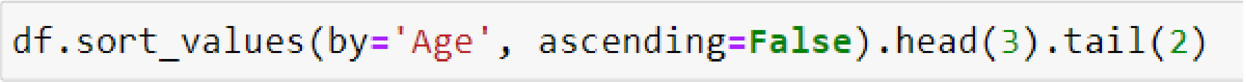
```python
import pandas as pd
```



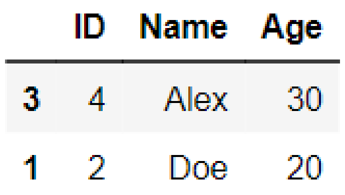
```python
data = {
    'ID': [1, 2, 3, 4, 5],
    'Name': ['Joe', 'Doe', 'Paula', 'Alex', 'John'],
    'Age': [10, 20, 40, 30, 15]
}

df = pd.DataFrame(data)

df
```



	ID	Name	Age
0	1	Joe	10
1	2	Doe	20
2	3	Paula	40
3	4	Alex	30
4	5	John	15



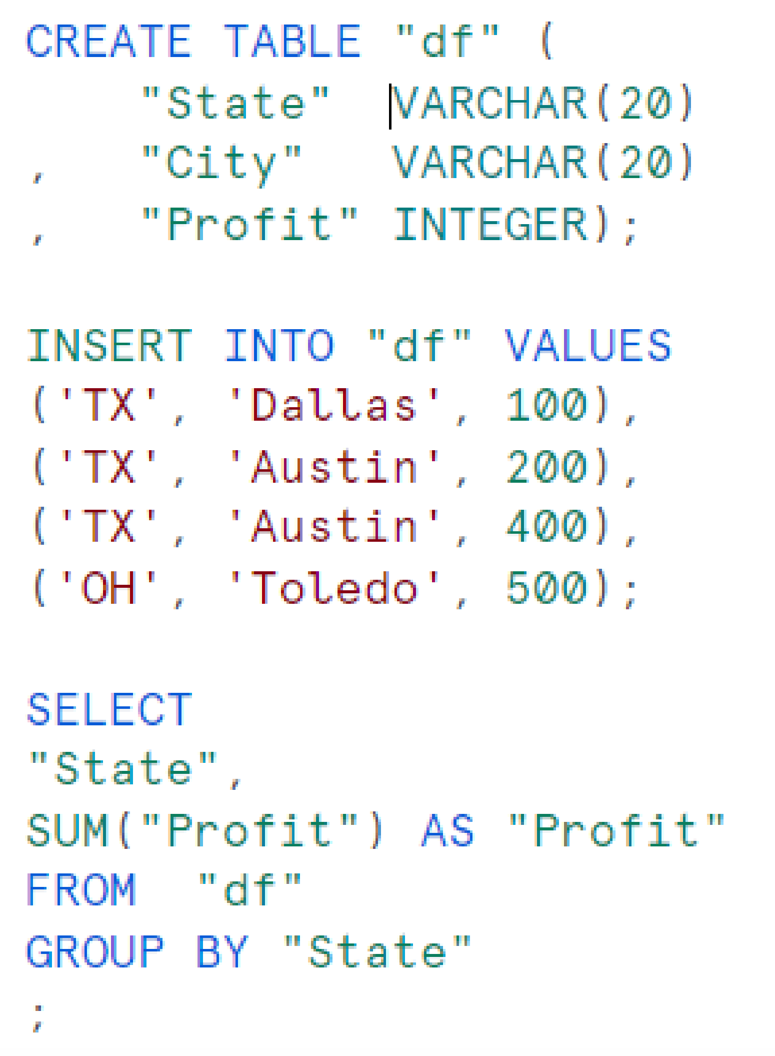
```python
df.sort_values(by='Age', ascending=False).head(3).tail(2)
```



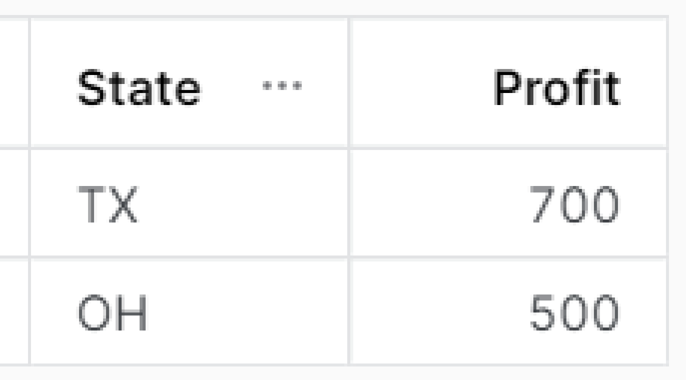
	ID	Name	Age
3	4	Alex	30
1	2	Doe	20

in Uzwal Goud Vaddeboina

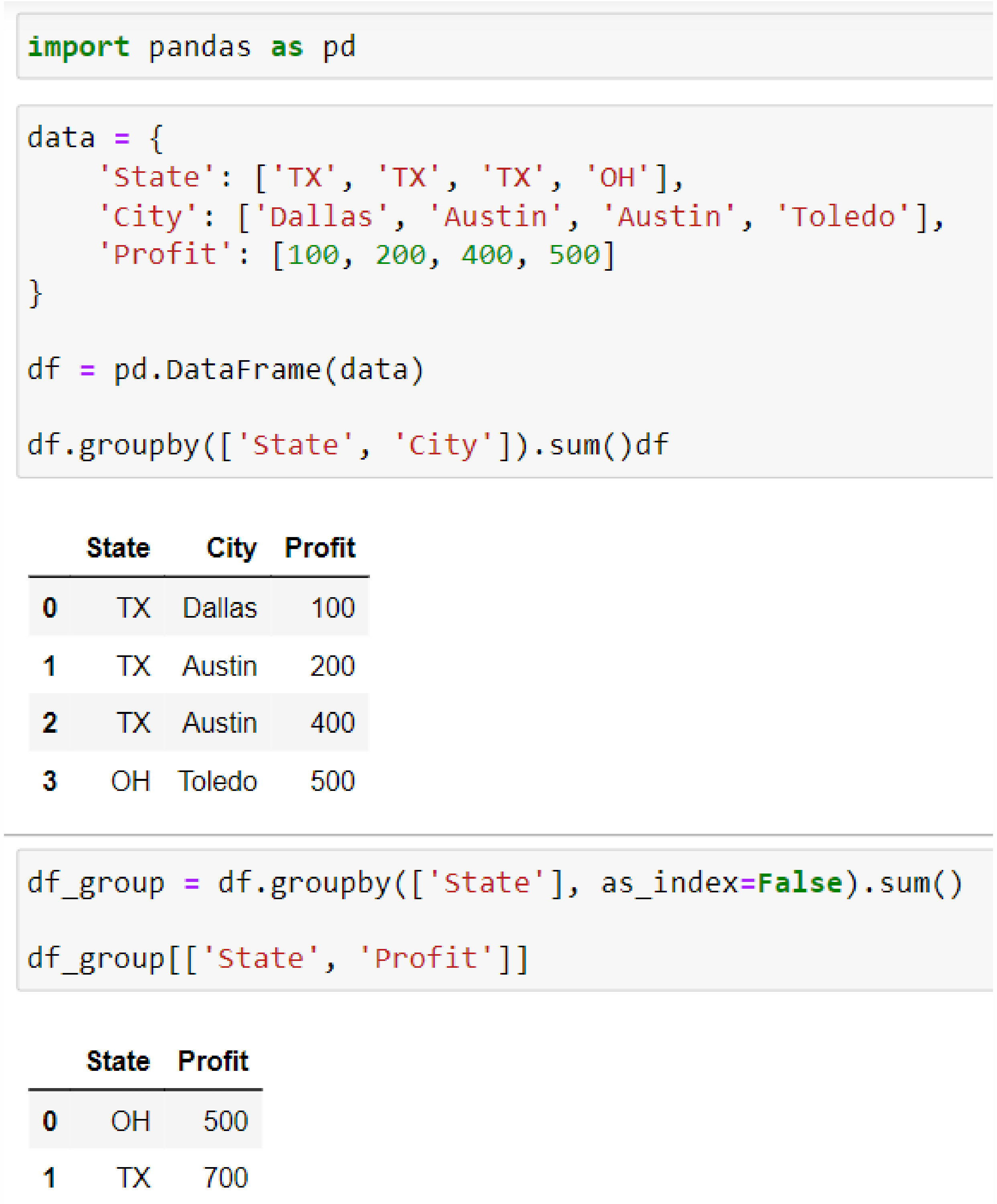
GROUP BY SINGLE COLUMN



```sql
CREATE TABLE "df" (
    "State"  VARCHAR(20)
,   "City"   VARCHAR(20)
,   "Profit" INTEGER);

INSERT INTO "df" VALUES
('TX', 'Dallas', 100),
('TX', 'Austin', 200),
('TX', 'Austin', 400),
('OH', 'Toledo', 500);

SELECT
"State",
SUM("Profit") AS "Profit"
FROM  "df"
GROUP BY "State"
;
```



State ···	Profit
TX	700
OH	500



```python
import pandas as pd
```

```python
data = {
    'State': ['TX', 'TX', 'TX', 'OH'],
    'City': ['Dallas', 'Austin', 'Austin', 'Toledo'],
    'Profit': [100, 200, 400, 500]
}

df = pd.DataFrame(data)

df.groupby(['State', 'City']).sum()df
```

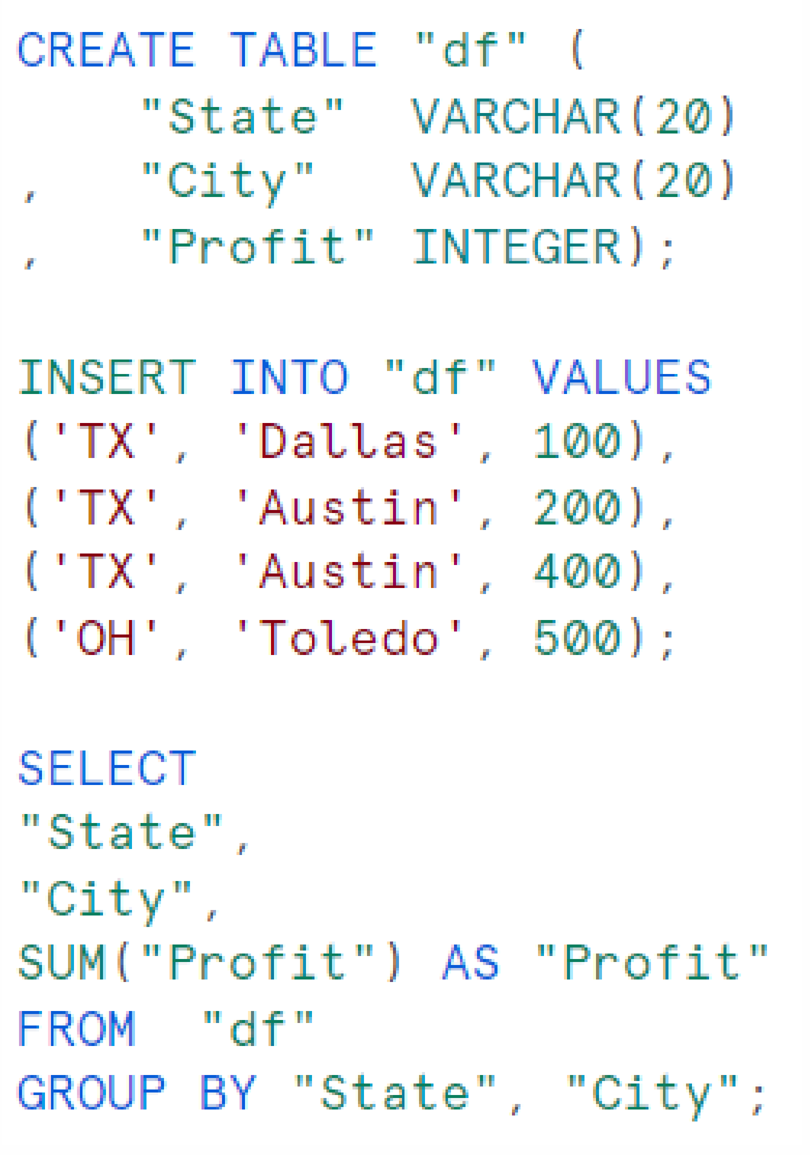
	State	City	Profit
0	TX	Dallas	100
1	TX	Austin	200
2	TX	Austin	400
3	OH	Toledo	500

```python
df_group = df.groupby(['State'], as_index=False).sum()

df_group[['State', 'Profit']]
```

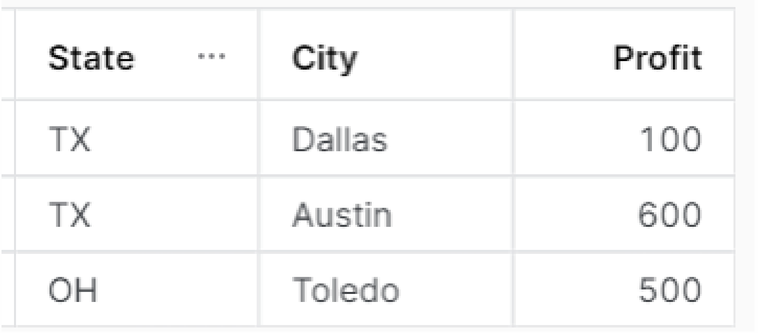
	State	Profit
0	OH	500
1	TX	700

Uzwal Goud Vaddeboina

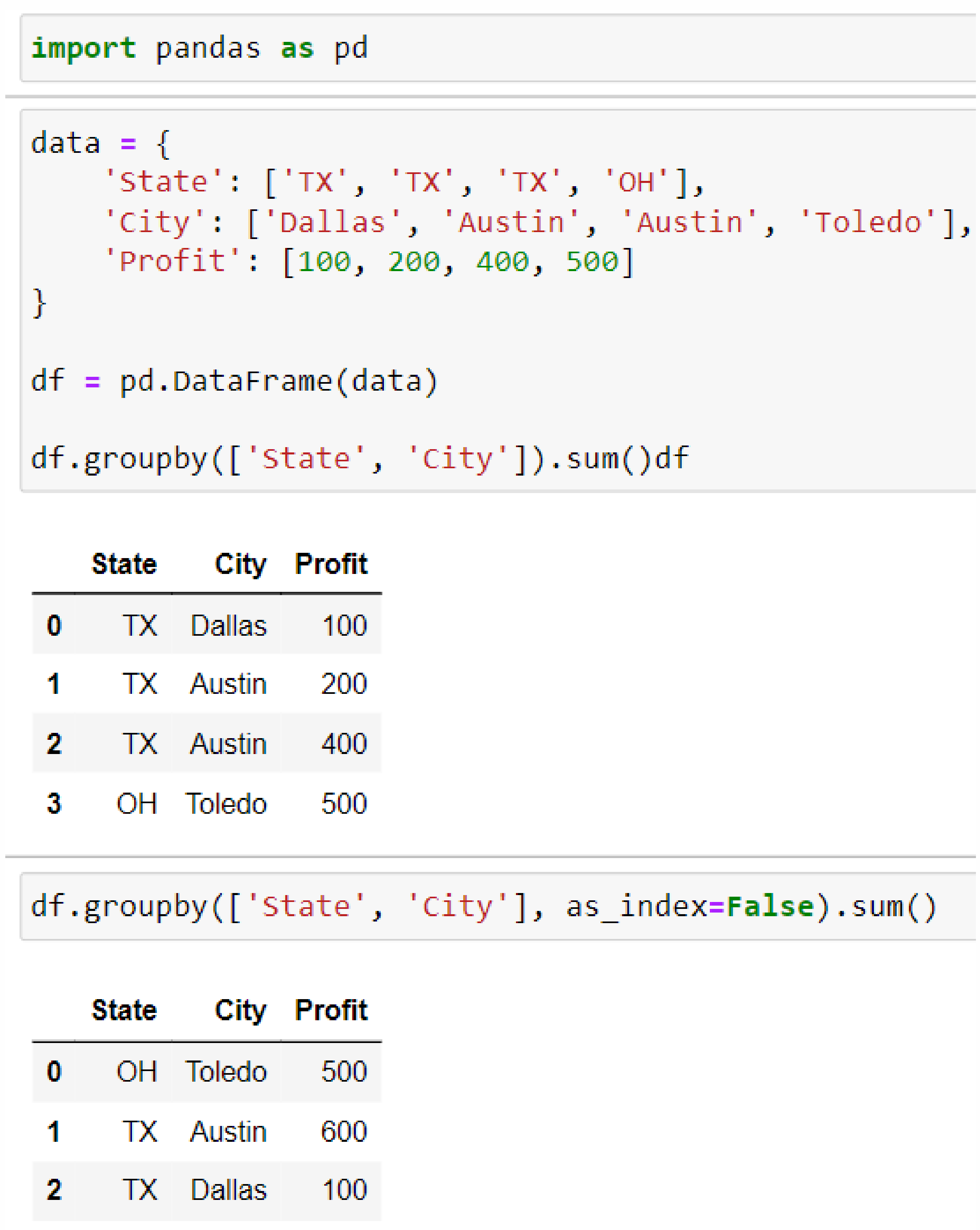
GROUP BY MULTIPLE COLUMNS



```sql
CREATE TABLE "df" (
    "State"  VARCHAR(20)
,   "City"   VARCHAR(20)
,   "Profit" INTEGER);

INSERT INTO "df" VALUES
('TX', 'Dallas', 100),
('TX', 'Austin', 200),
('TX', 'Austin', 400),
('OH', 'Toledo', 500);

SELECT
"State",
"City",
SUM("Profit") AS "Profit"
FROM  "df"
GROUP BY "State", "City";
```



State ···	City	Profit
TX	Dallas	100
TX	Austin	600
OH	Toledo	500



```python
import pandas as pd
```

```python
data = {
    'State': ['TX', 'TX', 'TX', 'OH'],
    'City': ['Dallas', 'Austin', 'Austin', 'Toledo'],
    'Profit': [100, 200, 400, 500]
}

df = pd.DataFrame(data)

df.groupby(['State', 'City']).sum()df
```

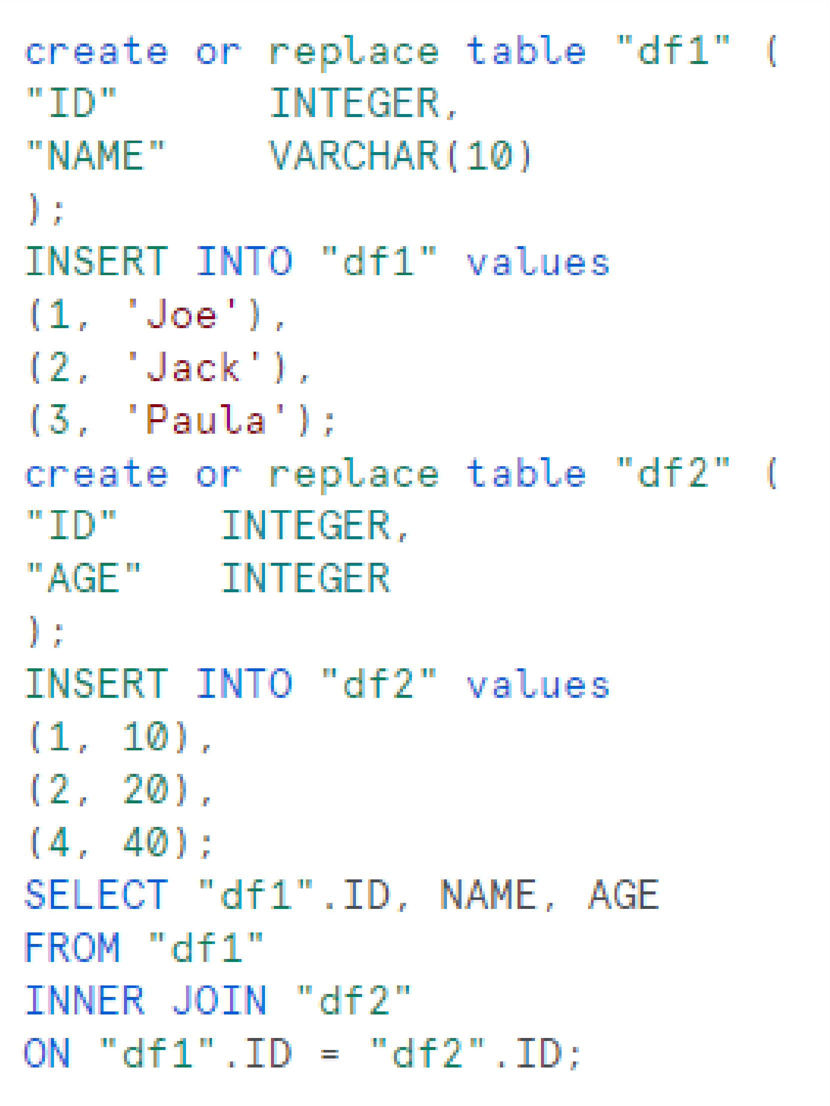
	State	City	Profit
0	TX	Dallas	100
1	TX	Austin	200
2	TX	Austin	400
3	OH	Toledo	500

```python
df.groupby(['State', 'City'], as_index=False).sum()
```

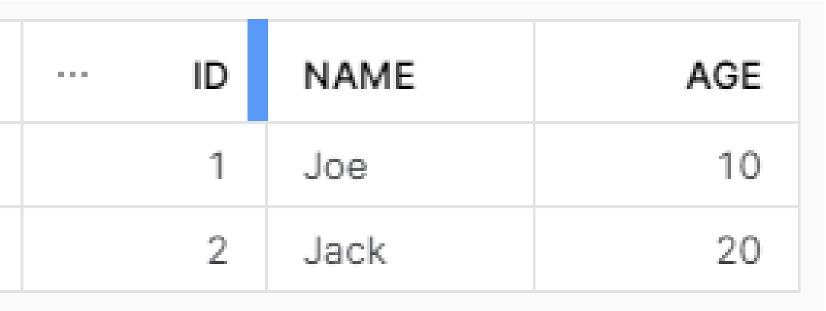
	State	City	Profit
0	OH	Toledo	500
1	TX	Austin	600
2	TX	Dallas	100

in Uzwal Goud Vaddeboina

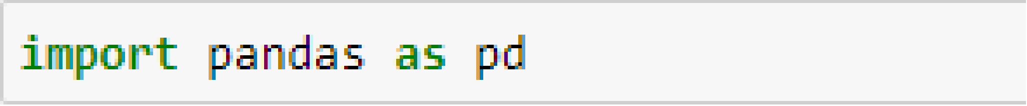
INNER JOIN



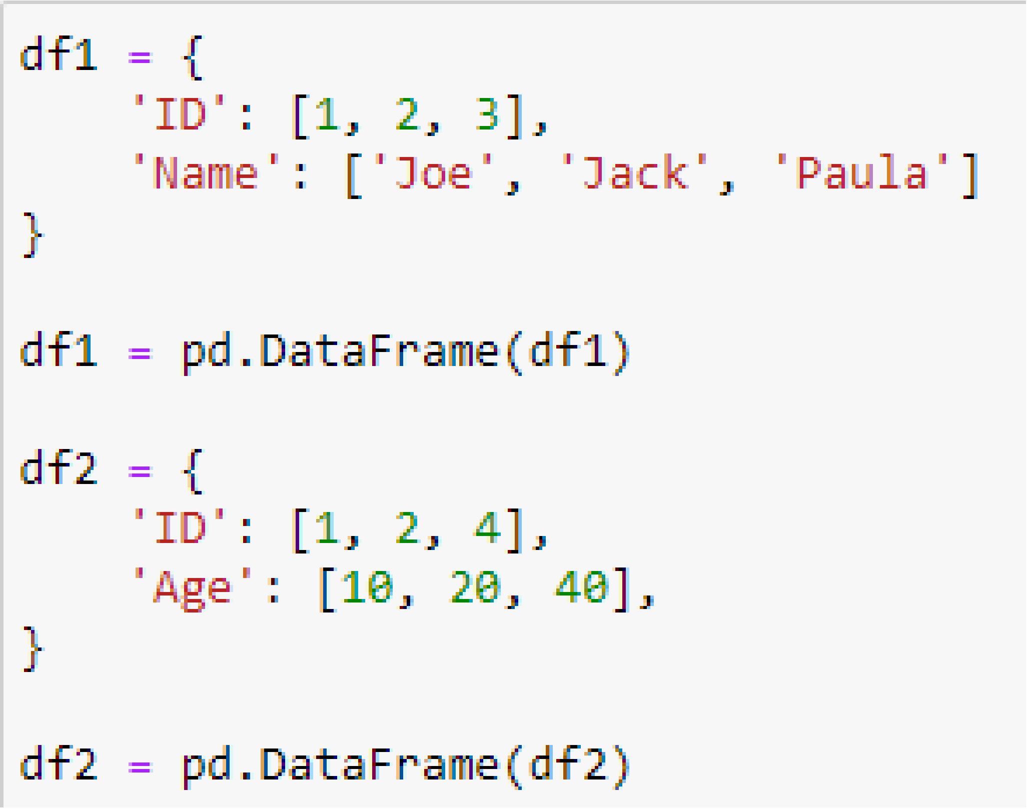
```sql
create or replace table "df1" (
"ID"      INTEGER,
"NAME"    VARCHAR(10)
);
INSERT INTO "df1" values
(1, 'Joe'),
(2, 'Jack'),
(3, 'Paula');
create or replace table "df2" (
"ID"    INTEGER,
"AGE"   INTEGER
);
INSERT INTO "df2" values
(1, 10),
(2, 20),
(4, 40);
SELECT "df1".ID, NAME, AGE
FROM "df1"
INNER JOIN "df2"
ON "df1".ID = "df2".ID;
```



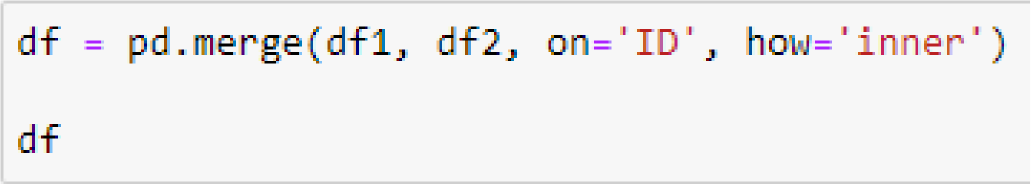
···	ID	NAME	AGE
	1	Joe	10
	2	Jack	20



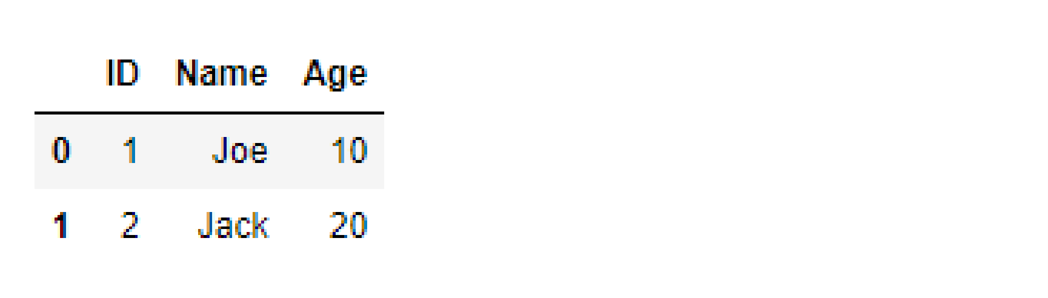
```python
import pandas as pd
```



```python
df1 = {
     'ID': [1, 2, 3],
     'Name': ['Joe', 'Jack', 'Paula']
}

df1 = pd.DataFrame(df1)

df2 = {
     'ID': [1, 2, 4],
     'Age': [10, 20, 40],
}

df2 = pd.DataFrame(df2)
```



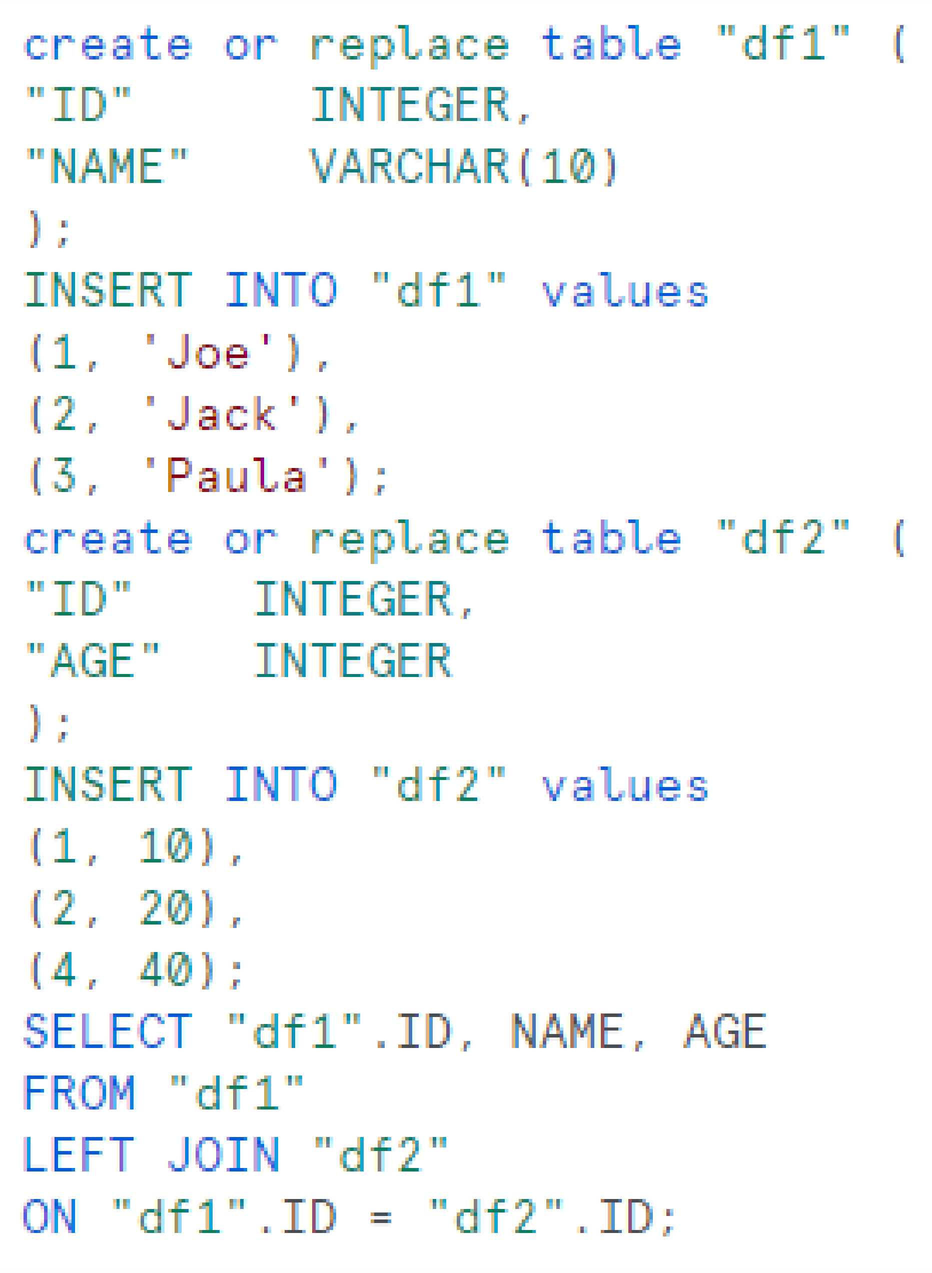
```python
df = pd.merge(df1, df2, on='ID', how='inner')

df
```



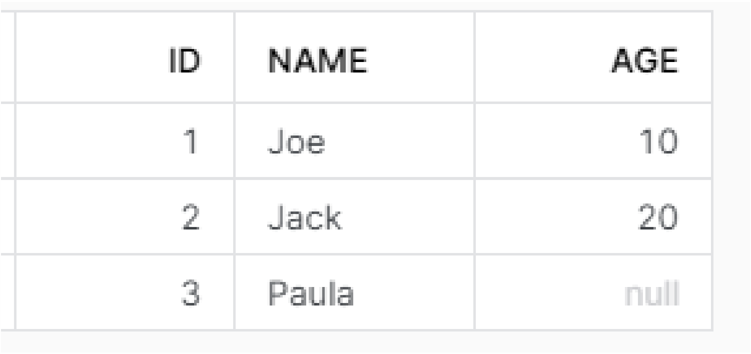
	ID	Name	Age
0	1	Joe	10
1	2	Jack	20

Uzwal Goud Vaddeboina

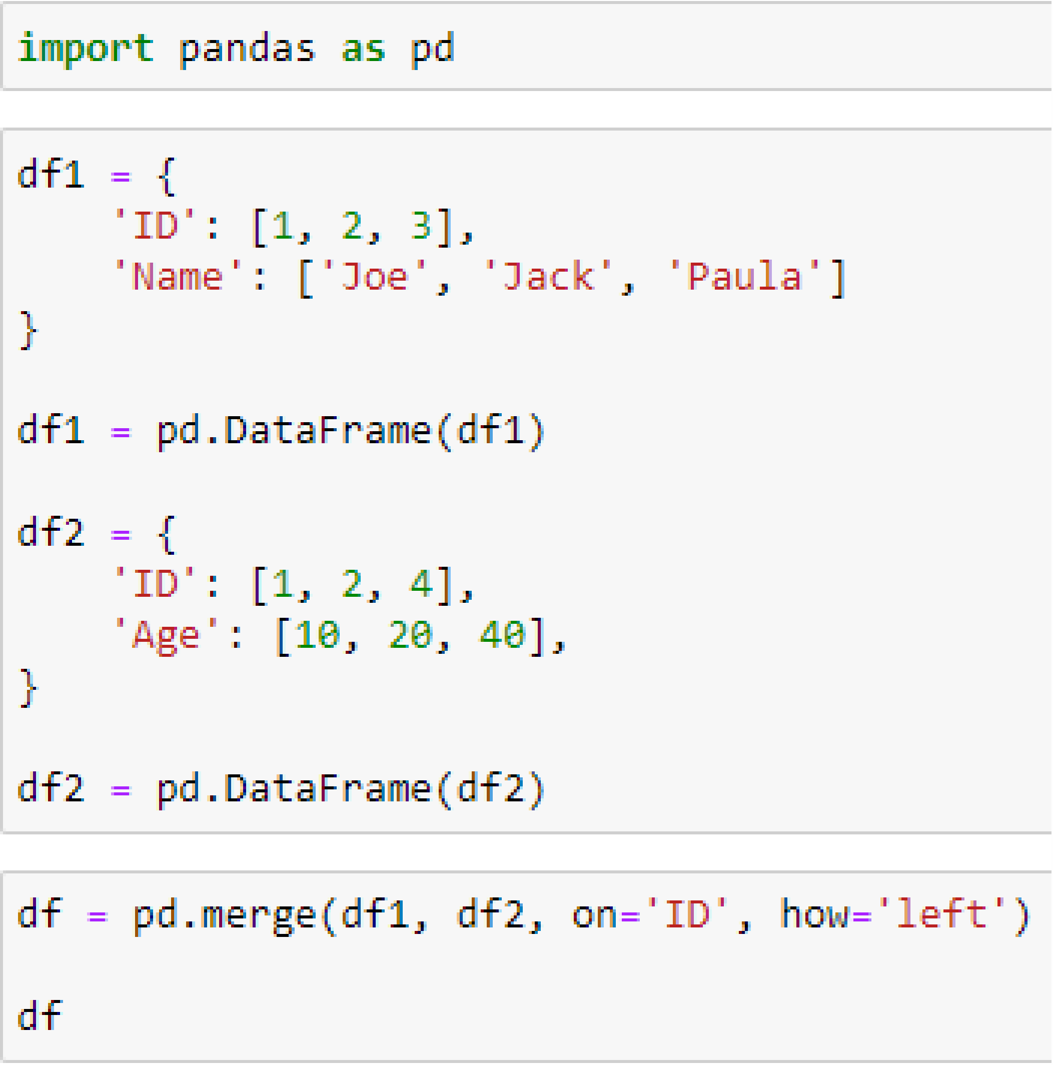
LEFT JOIN



```
create or replace table "df1" (
"ID"        INTEGER,
"NAME"      VARCHAR(10)
);
INSERT INTO "df1" values
(1, 'Joe'),
(2, 'Jack'),
(3, 'Paula');
create or replace table "df2" (
"ID"      INTEGER,
"AGE"     INTEGER
);
INSERT INTO "df2" values
(1, 10),
(2, 20),
(4, 40);
SELECT "df1".ID, NAME, AGE
FROM "df1"
LEFT JOIN "df2"
ON "df1".ID = "df2".ID;
```



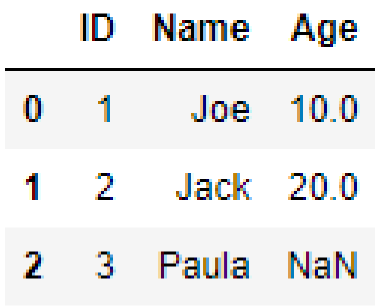
ID	NAME	AGE
1	Joe	10
2	Jack	20
3	Paula	null



```python
import pandas as pd
```

```python
df1 = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Jack', 'Paula']
}

df1 = pd.DataFrame(df1)

df2 = {
    'ID': [1, 2, 4],
    'Age': [10, 20, 40],
}

df2 = pd.DataFrame(df2)
```

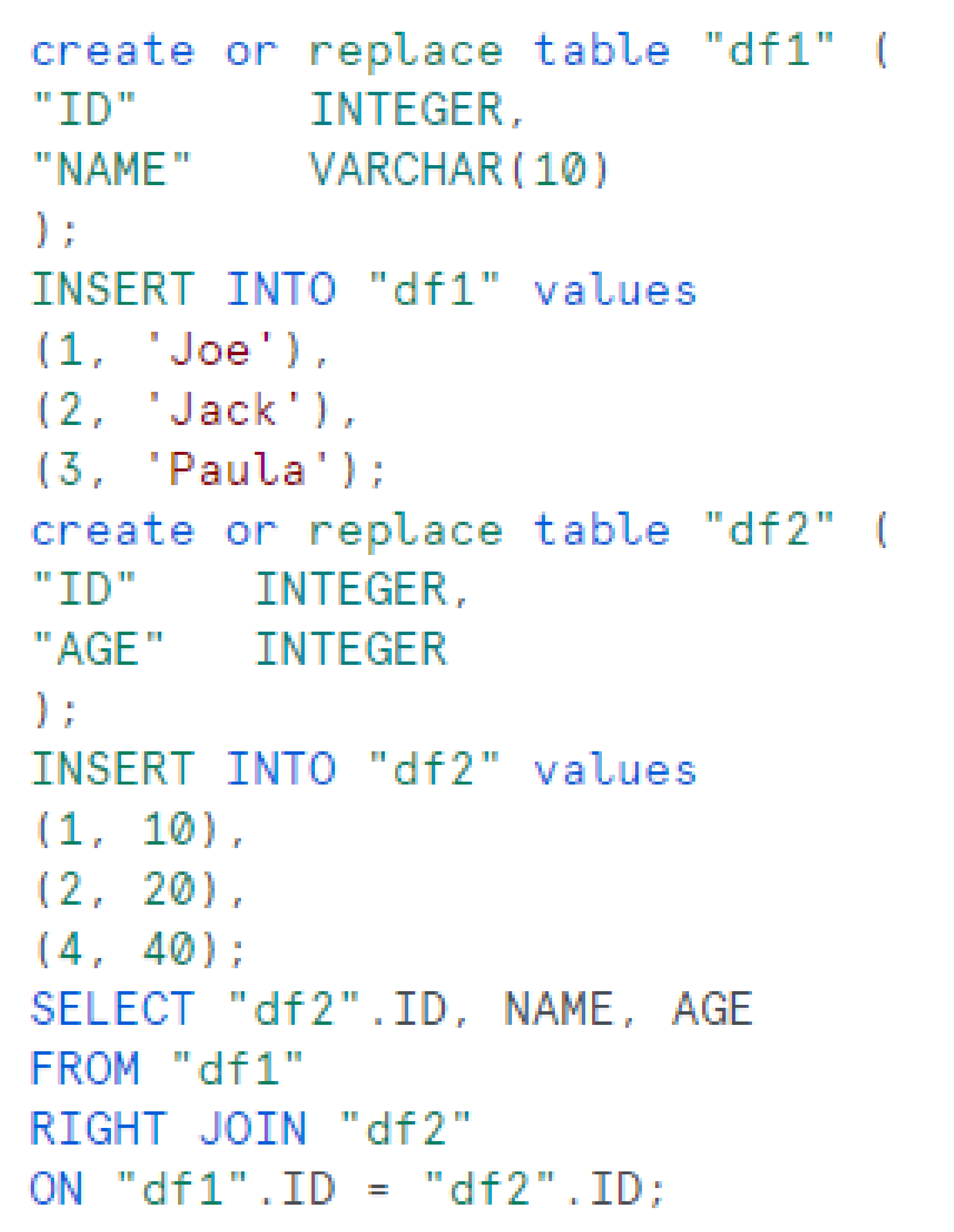
```python
df = pd.merge(df1, df2, on='ID', how='left')

df
```



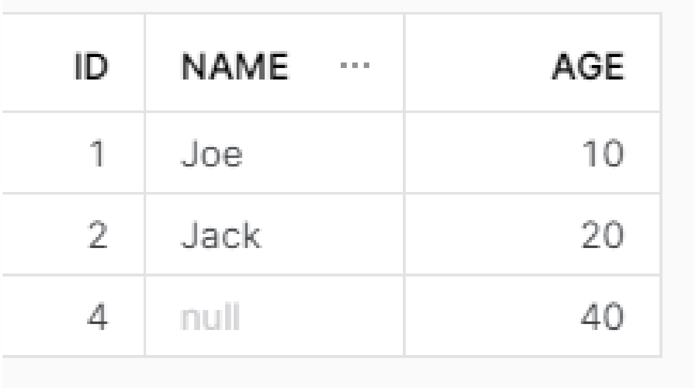
	ID	Name	Age
0	1	Joe	10.0
1	2	Jack	20.0
2	3	Paula	NaN

Uzwal Goud Vaddeboina

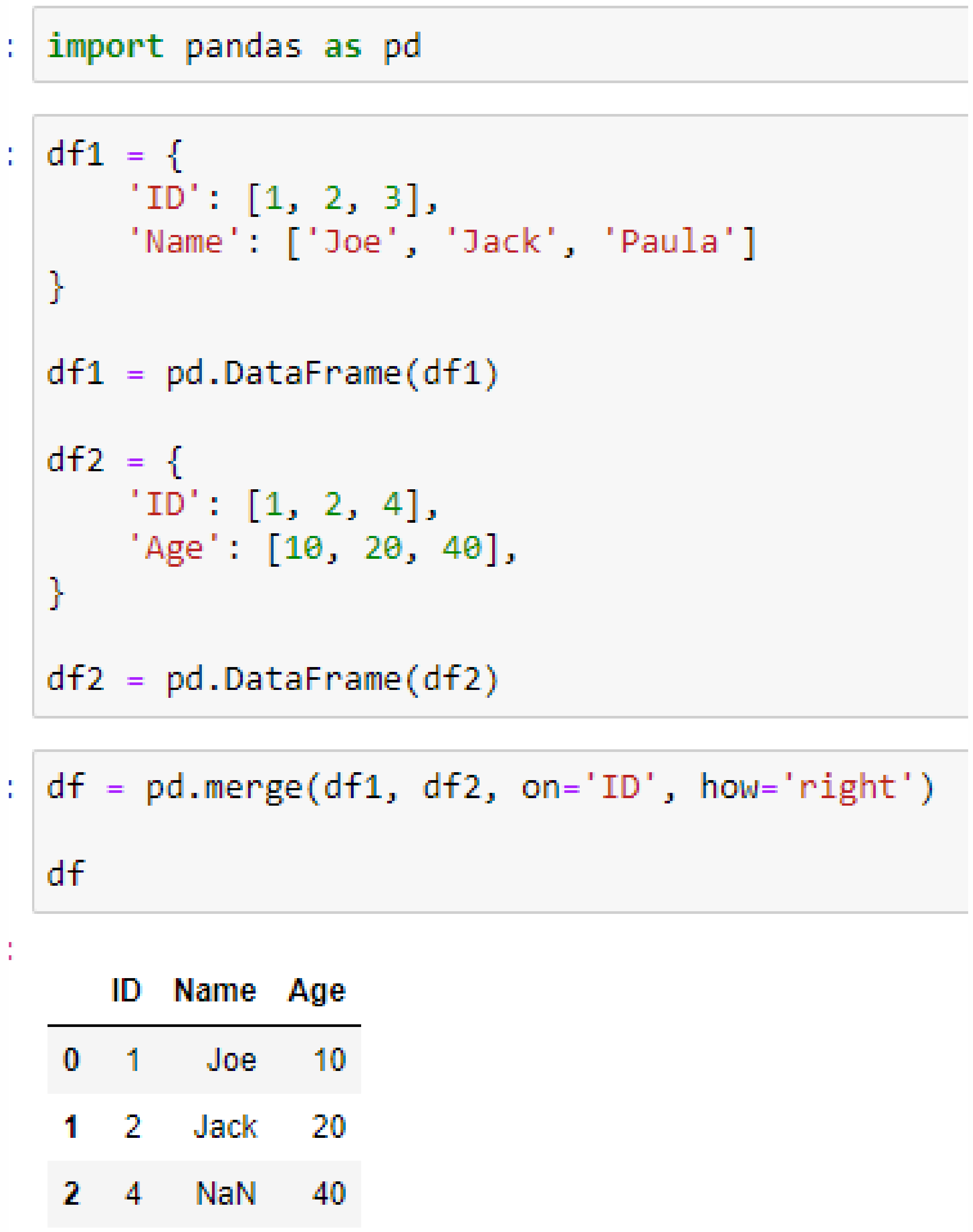
RIGHT JOIN



```sql
create or replace table "df1" (
"ID"       INTEGER,
"NAME"     VARCHAR(10)
);
INSERT INTO "df1" values
(1, 'Joe'),
(2, 'Jack'),
(3, 'Paula');
create or replace table "df2" (
"ID"     INTEGER,
"AGE"    INTEGER
);
INSERT INTO "df2" values
(1, 10),
(2, 20),
(4, 40);
SELECT "df2".ID, NAME, AGE
FROM "df1"
RIGHT JOIN "df2"
ON "df1".ID = "df2".ID;
```



ID	NAME ···	AGE
1	Joe	10
2	Jack	20
4	null	40



```python
import pandas as pd
```

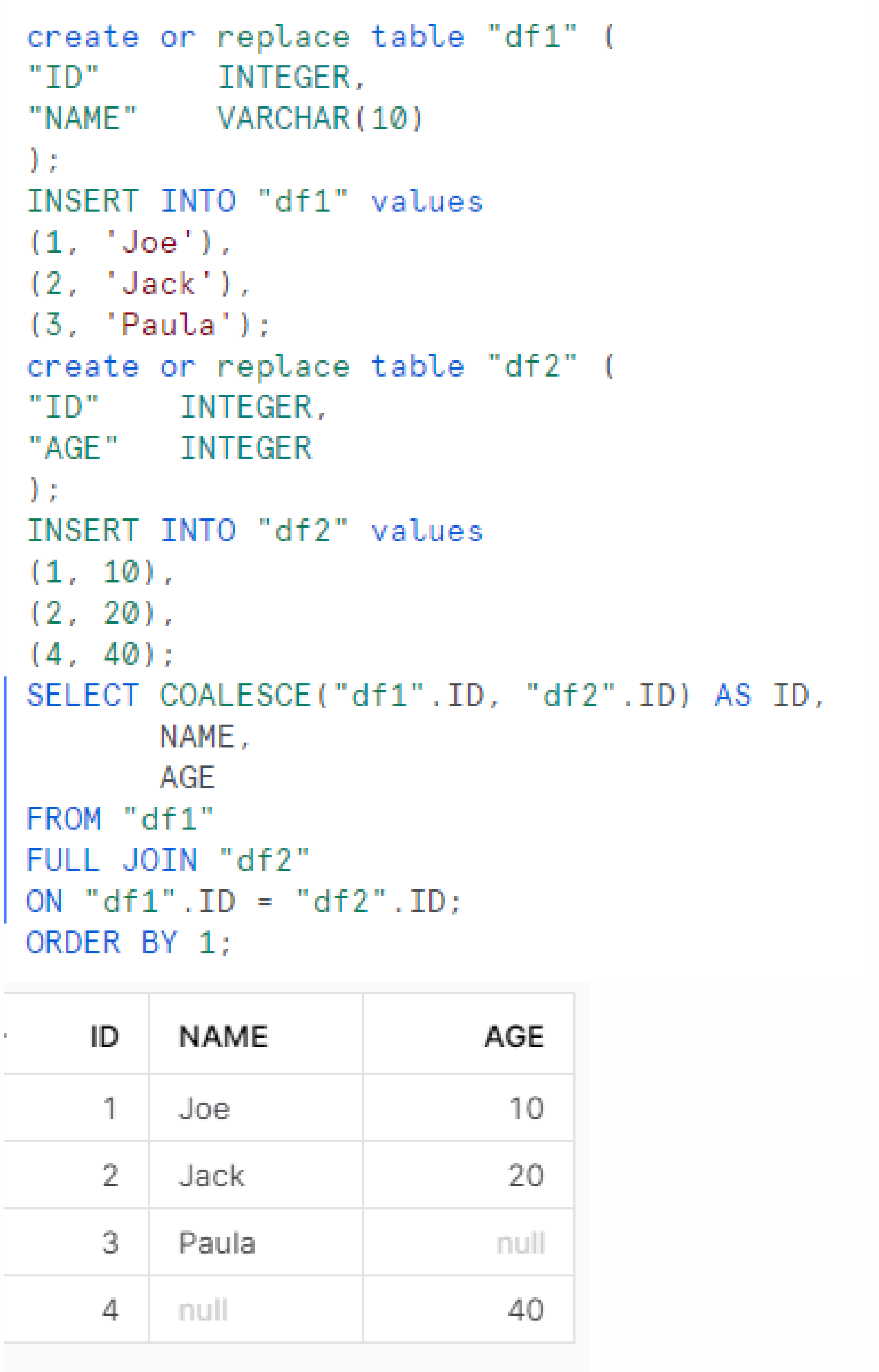
```python
df1 = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Jack', 'Paula']
}

df1 = pd.DataFrame(df1)

df2 = {
    'ID': [1, 2, 4],
    'Age': [10, 20, 40],
}

df2 = pd.DataFrame(df2)
```

```python
df = pd.merge(df1, df2, on='ID', how='right')

df
```

	ID	Name	Age
0	1	Joe	10
1	2	Jack	20
2	4	NaN	40

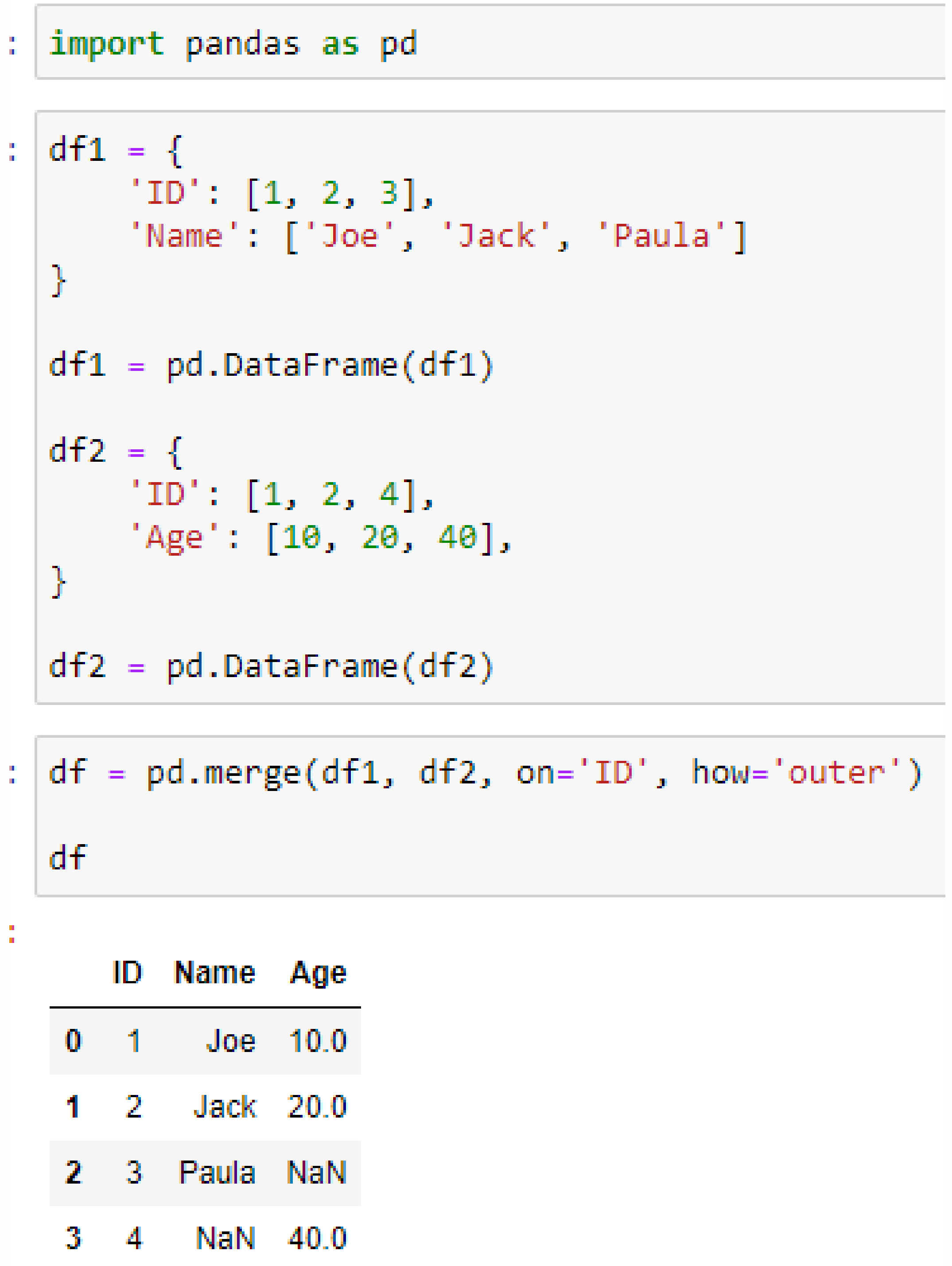
Uzwal Goud Vaddeboina

FULL JOIN



```sql
create or replace table "df1" (
"ID"       INTEGER,
"NAME"     VARCHAR(10)
);
INSERT INTO "df1" values
(1, 'Joe'),
(2, 'Jack'),
(3, 'Paula');
create or replace table "df2" (
"ID"     INTEGER,
"AGE"    INTEGER
);
INSERT INTO "df2" values
(1, 10),
(2, 20),
(4, 40);
SELECT COALESCE("df1".ID, "df2".ID) AS ID,
       NAME,
       AGE
FROM "df1"
FULL JOIN "df2"
ON "df1".ID = "df2".ID;
ORDER BY 1;
```

ID	NAME	AGE
1	Joe	10
2	Jack	20
3	Paula	null
4	null	40



```python
import pandas as pd
```

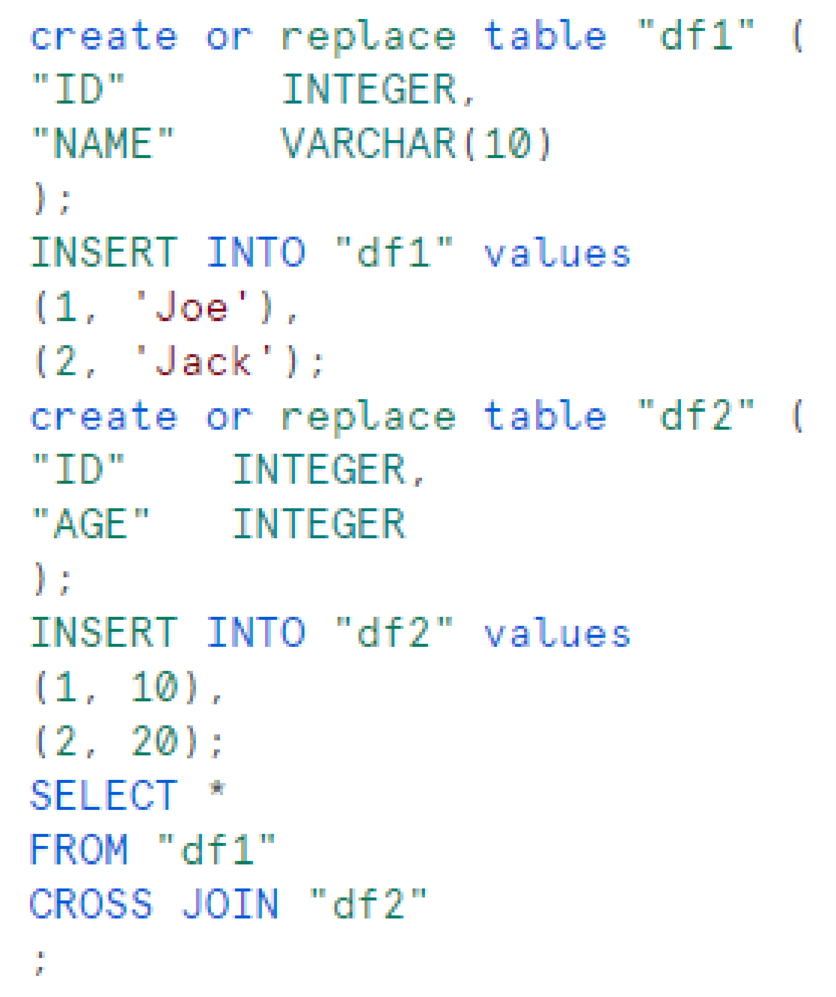
```python
df1 = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Jack', 'Paula']
}

df1 = pd.DataFrame(df1)

df2 = {
    'ID': [1, 2, 4],
    'Age': [10, 20, 40],
}

df2 = pd.DataFrame(df2)
```

```python
df = pd.merge(df1, df2, on='ID', how='outer')

df
```

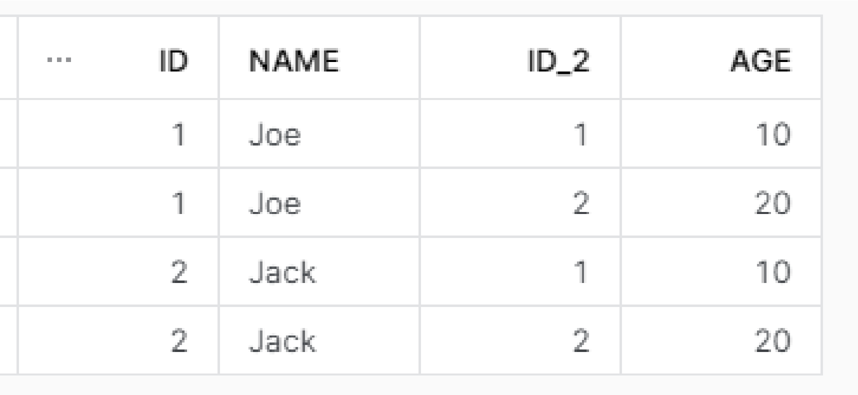
	ID	Name	Age
0	1	Joe	10.0
1	2	Jack	20.0
2	3	Paula	NaN
3	4	NaN	40.0

Uzwal Goud Vaddeboina

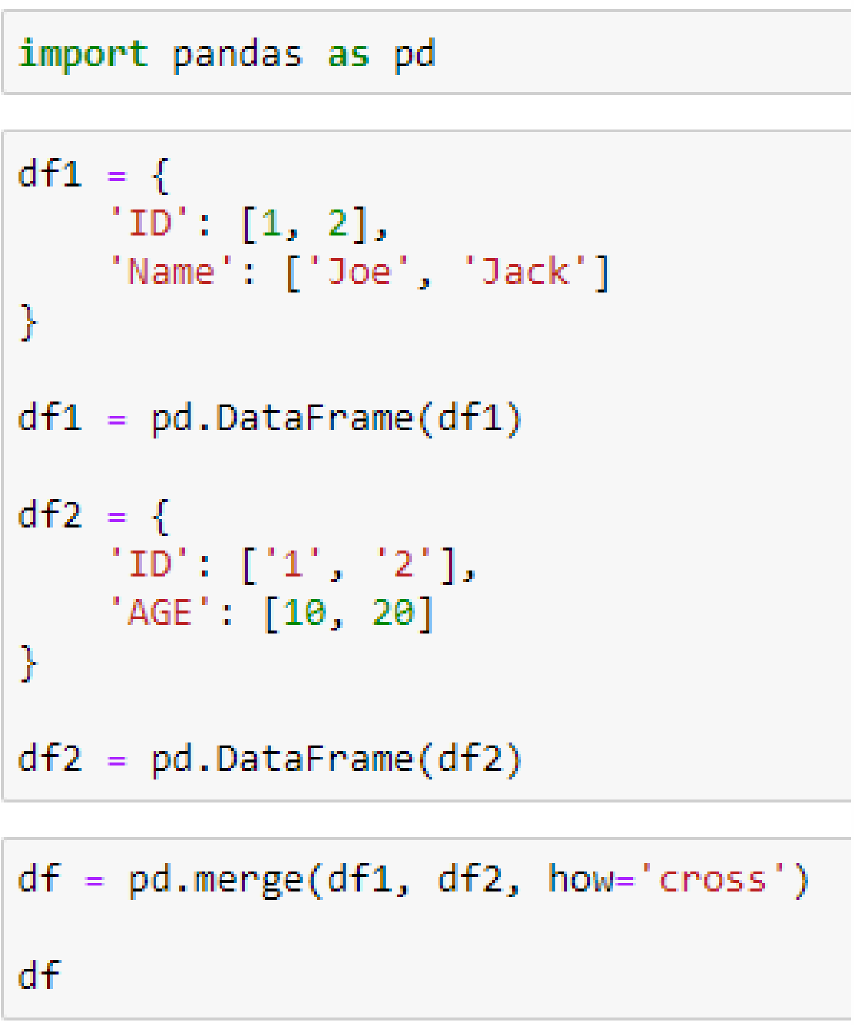
CROSS JOIN



```sql
create or replace table "df1" (
"ID"       INTEGER,
"NAME"     VARCHAR(10)
);
INSERT INTO "df1" values
(1, 'Joe'),
(2, 'Jack');
create or replace table "df2" (
"ID"     INTEGER,
"AGE"    INTEGER
);
INSERT INTO "df2" values
(1, 10),
(2, 20);
SELECT *
FROM "df1"
CROSS JOIN "df2"
;
```



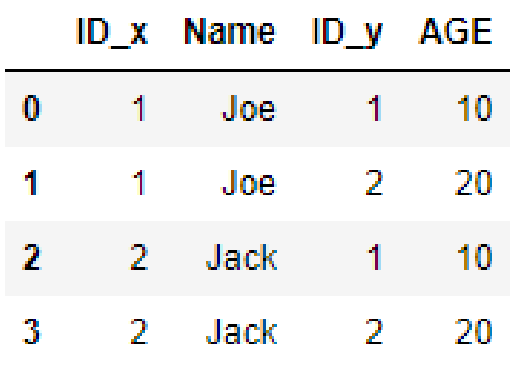
···	ID	NAME	ID_2	AGE
	1	Joe	1	10
	1	Joe	2	20
	2	Jack	1	10
	2	Jack	2	20



```python
import pandas as pd
```

```python
df1 = {
    'ID': [1, 2],
    'Name': ['Joe', 'Jack']
}

df1 = pd.DataFrame(df1)

df2 = {
    'ID': ['1', '2'],
    'AGE': [10, 20]
}

df2 = pd.DataFrame(df2)
```

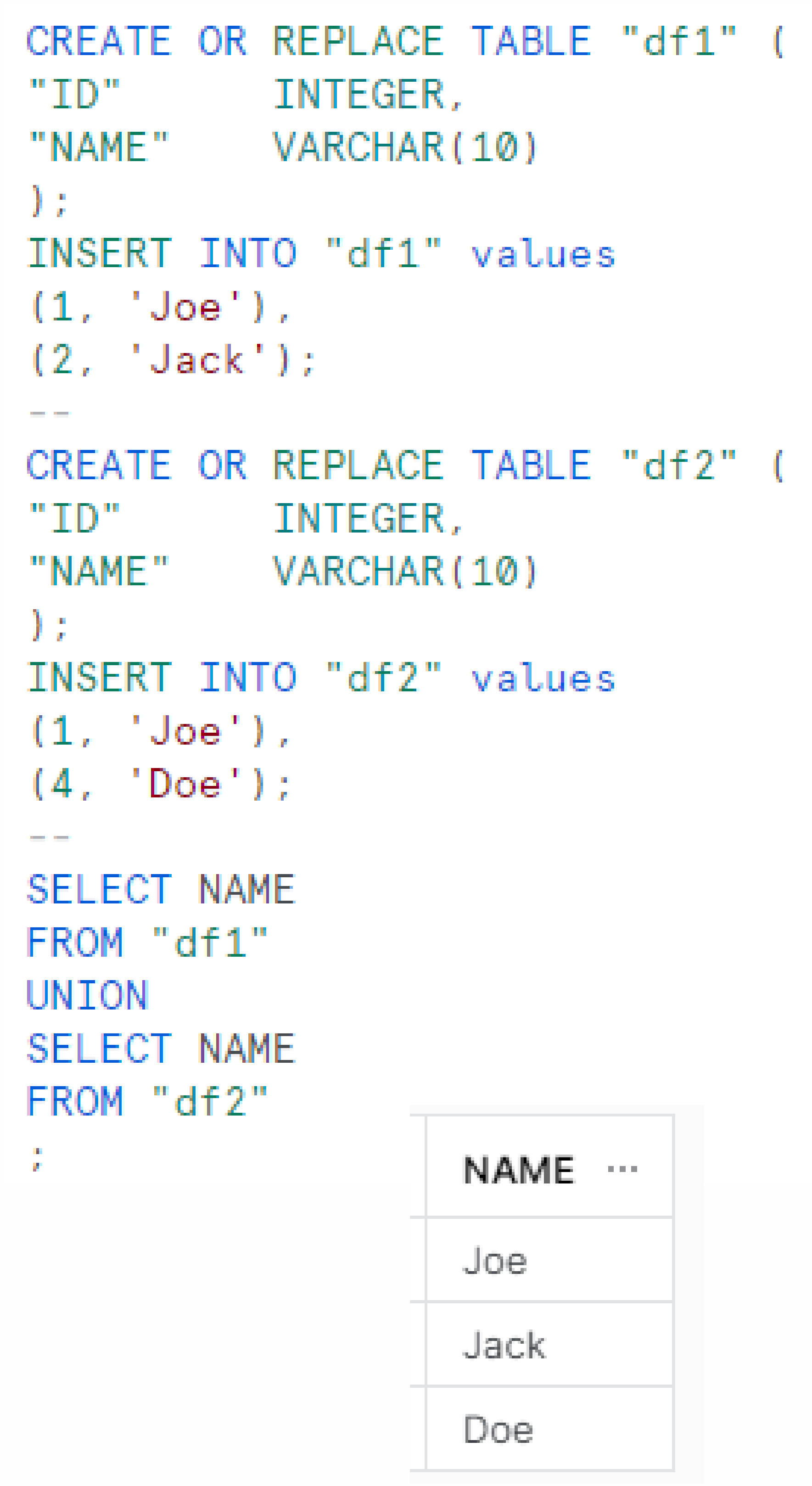
```python
df = pd.merge(df1, df2, how='cross')

df
```



	ID_x	Name	ID_y	AGE
0	1	Joe	1	10
1	1	Joe	2	20
2	2	Jack	1	10
3	2	Jack	2	20

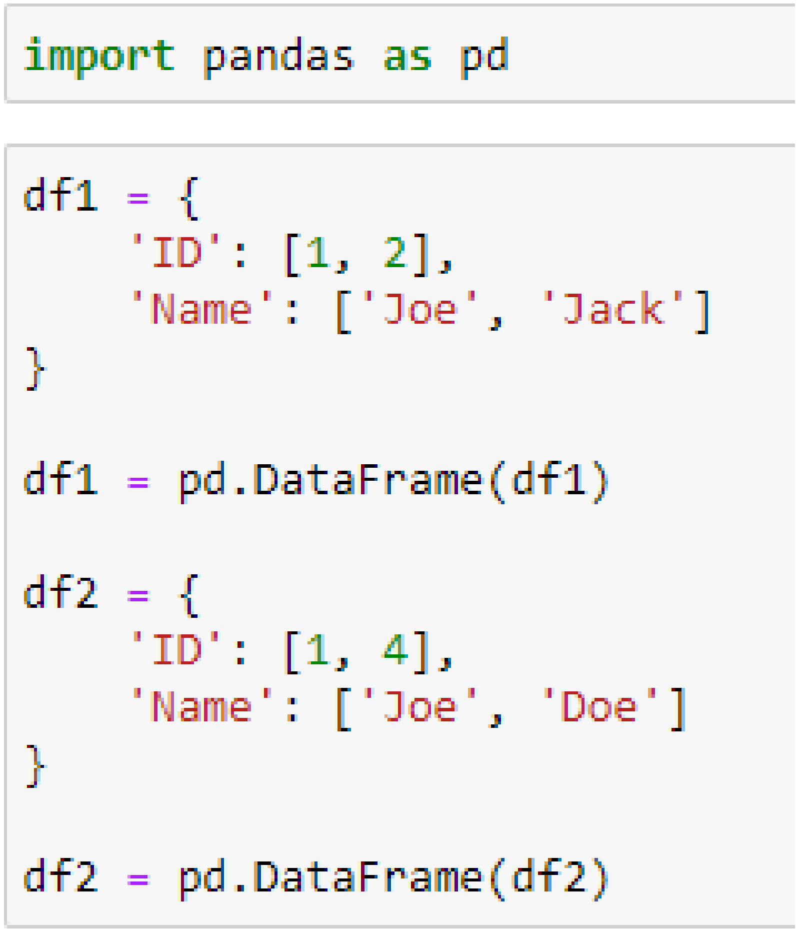
Uzwal Goud Vaddeboina

UNION BY SINGLE COLUMN



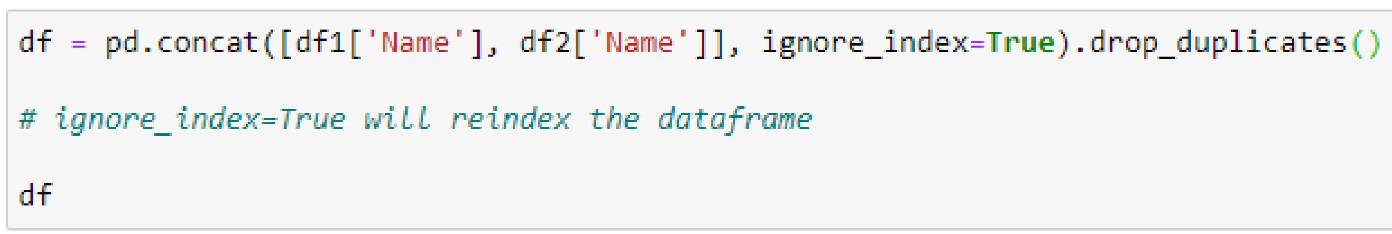
```sql
CREATE OR REPLACE TABLE "df1" (
"ID"        INTEGER,
"NAME"      VARCHAR(10)
);
INSERT INTO "df1" values
(1, 'Joe'),
(2, 'Jack');
--
CREATE OR REPLACE TABLE "df2" (
"ID"        INTEGER,
"NAME"      VARCHAR(10)
);
INSERT INTO "df2" values
(1, 'Joe'),
(4, 'Doe');
--
SELECT NAME
FROM "df1"
UNION
SELECT NAME
FROM "df2"
;
```

NAME ···
Joe
Jack
Doe

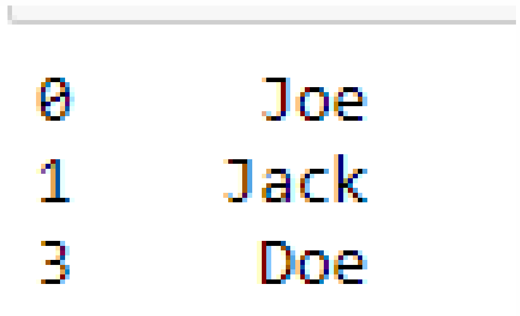


```python
import pandas as pd
```

```python
df1 = {
     'ID': [1, 2],
     'Name': ['Joe', 'Jack']
}

df1 = pd.DataFrame(df1)

df2 = {
     'ID': [1, 4],
     'Name': ['Joe', 'Doe']
}

df2 = pd.DataFrame(df2)
```



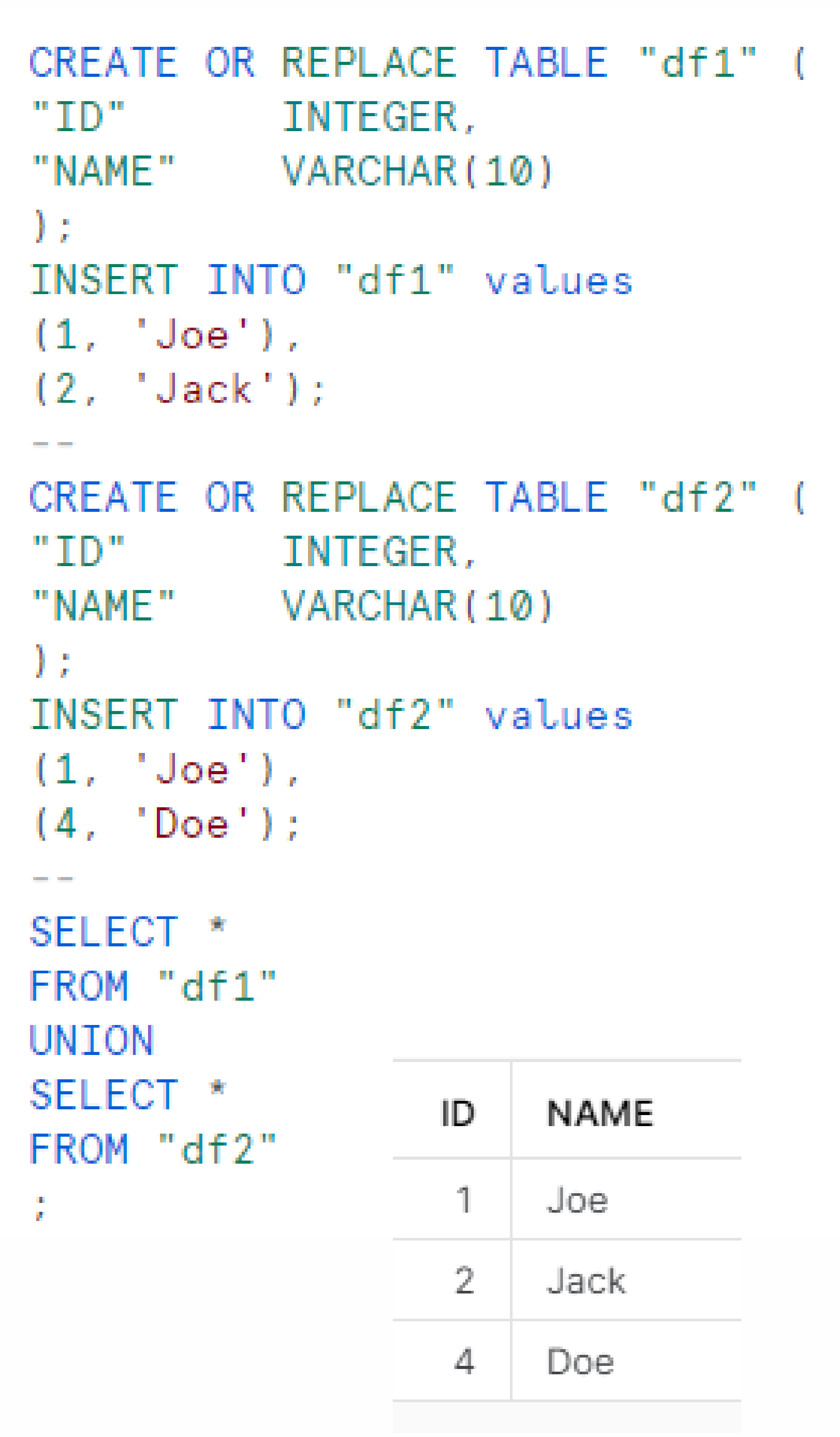
```python
df = pd.concat([df1['Name'], df2['Name']], ignore_index=True).drop_duplicates()

# ignore_index=True will reindex the dataframe

df
```



```
0       Joe
1      Jack
3       Doe
```

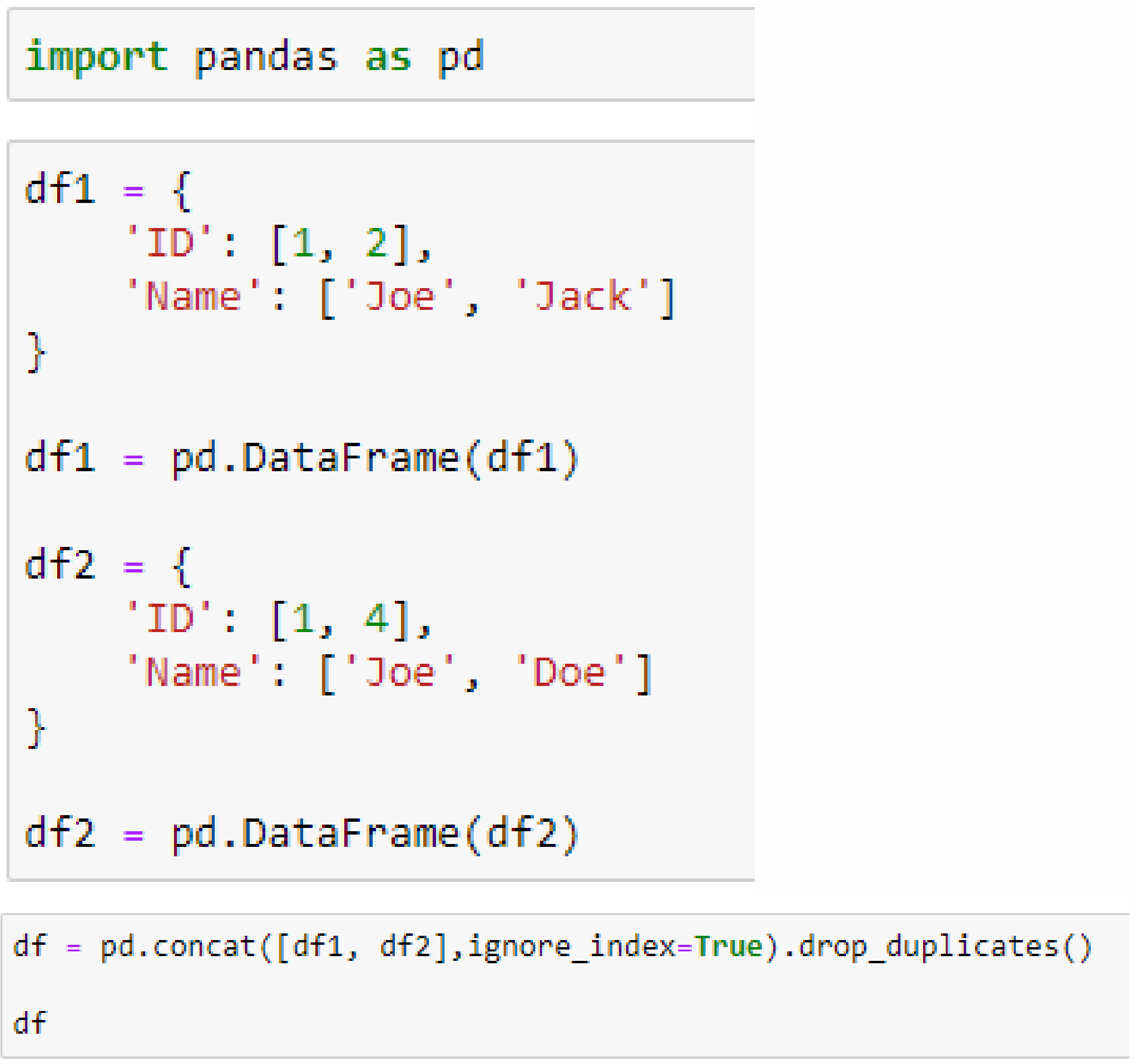
in Uzwal Goud Vaddeboina

UNION BY ALL COLUMNS



```sql
CREATE OR REPLACE TABLE "df1" (
"ID"        INTEGER,
"NAME"      VARCHAR(10)
);
INSERT INTO "df1" values
(1, 'Joe'),
(2, 'Jack');
--
CREATE OR REPLACE TABLE "df2" (
"ID"        INTEGER,
"NAME"      VARCHAR(10)
);
INSERT INTO "df2" values
(1, 'Joe'),
(4, 'Doe');
--
SELECT *
FROM "df1"
UNION
SELECT *
FROM "df2"
;
```

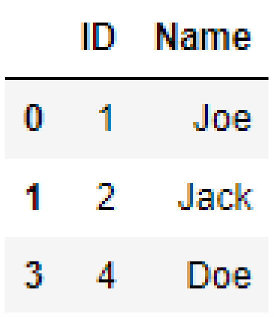
ID	NAME
1	Joe
2	Jack
4	Doe



```python
import pandas as pd
```

```python
df1 = {
      'ID': [1, 2],
      'Name': ['Joe', 'Jack']
}

df1 = pd.DataFrame(df1)

df2 = {
      'ID': [1, 4],
      'Name': ['Joe', 'Doe']
}

df2 = pd.DataFrame(df2)
```

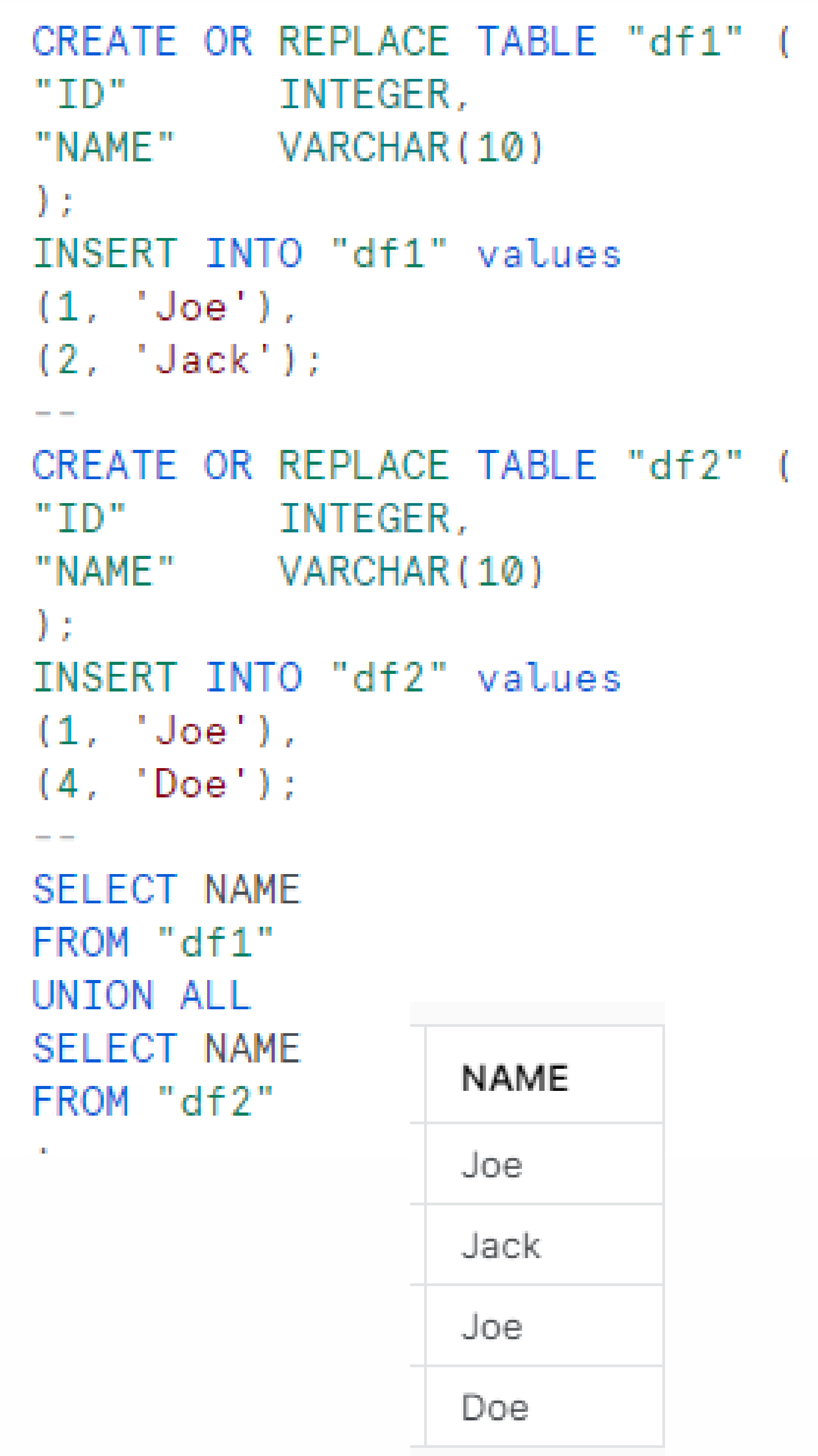
```python
df = pd.concat([df1, df2],ignore_index=True).drop_duplicates()

df
```



	ID	Name
0	1	Joe
1	2	Jack
3	4	Doe

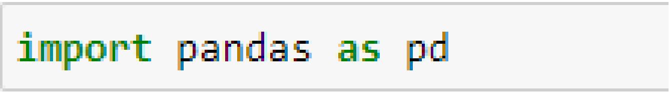
Uzwal Goud Vaddeboina

UNION ALL BY SINGLE COLUMN

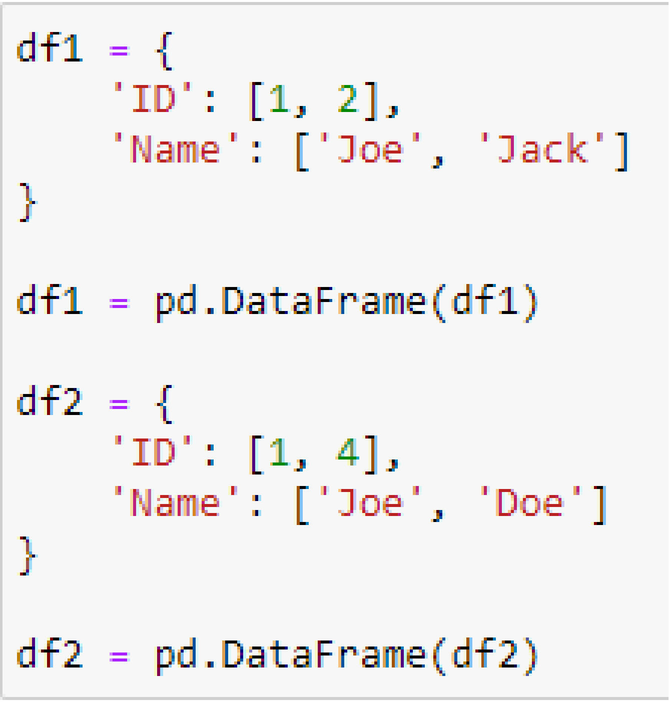


```sql
CREATE OR REPLACE TABLE "df1" (
"ID"        INTEGER,
"NAME"      VARCHAR(10)
);
INSERT INTO "df1" values
(1, 'Joe'),
(2, 'Jack');
--
CREATE OR REPLACE TABLE "df2" (
"ID"        INTEGER,
"NAME"      VARCHAR(10)
);
INSERT INTO "df2" values
(1, 'Joe'),
(4, 'Doe');
--
SELECT NAME
FROM "df1"
UNION ALL
SELECT NAME
FROM "df2"
.
```

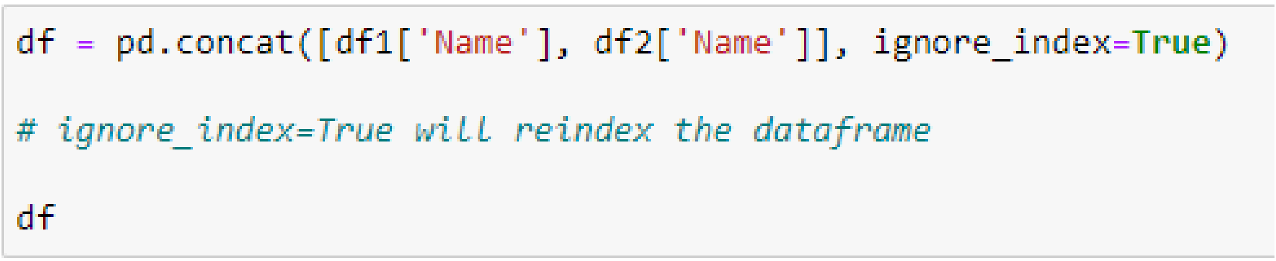
NAME
Joe
Jack
Joe
Doe



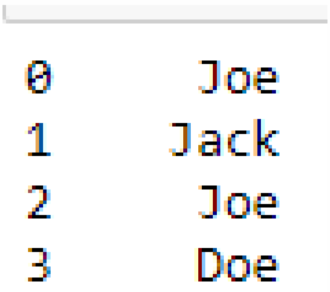
```python
import pandas as pd
```



```python
df1 = {
    'ID': [1, 2],
    'Name': ['Joe', 'Jack']
}

df1 = pd.DataFrame(df1)

df2 = {
    'ID': [1, 4],
    'Name': ['Joe', 'Doe']
}

df2 = pd.DataFrame(df2)
```



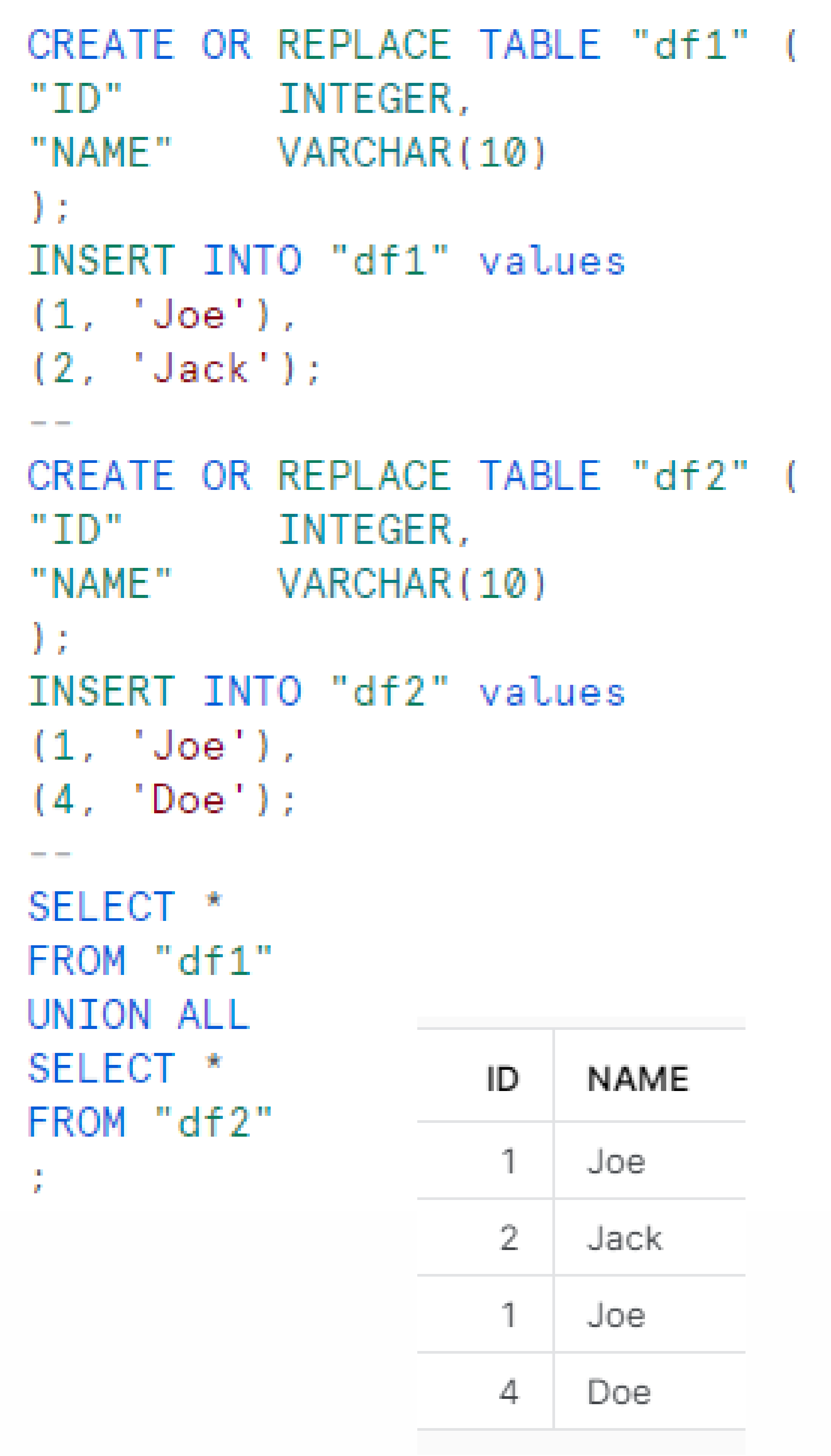
```python
df = pd.concat([df1['Name'], df2['Name']], ignore_index=True)

# ignore_index=True will reindex the dataframe

df
```



```
0     Joe
1    Jack
2     Joe
3     Doe
```

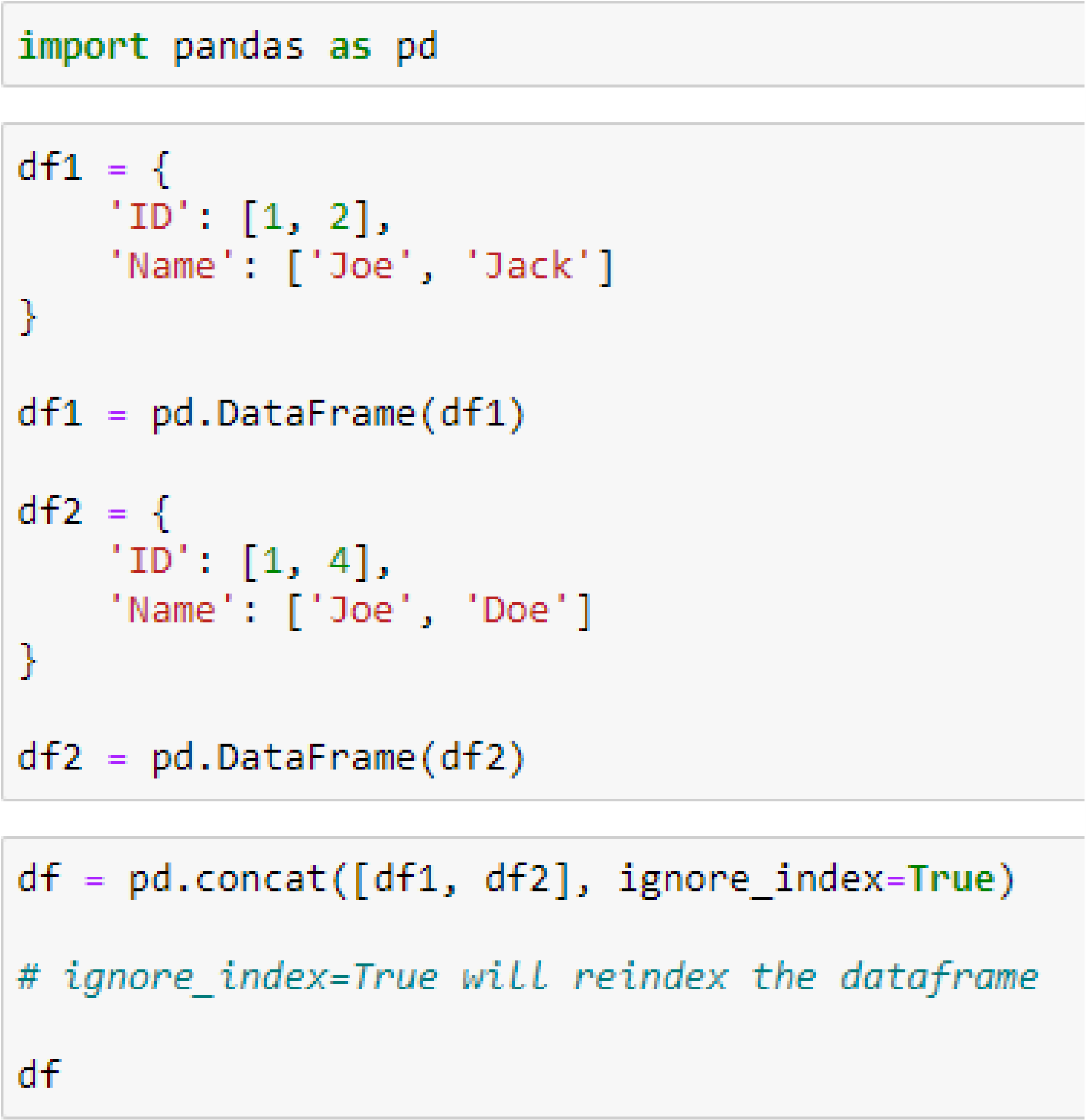
Uzwal Goud Vaddeboina

UNION ALL BY ALL COLUMNS



```
CREATE OR REPLACE TABLE "df1" (
"ID"        INTEGER,
"NAME"      VARCHAR(10)
);
INSERT INTO "df1" values
(1, 'Joe'),
(2, 'Jack');
--
CREATE OR REPLACE TABLE "df2" (
"ID"        INTEGER,
"NAME"      VARCHAR(10)
);
INSERT INTO "df2" values
(1, 'Joe'),
(4, 'Doe');
--
SELECT *
FROM "df1"
UNION ALL
SELECT *
FROM "df2"
;
```

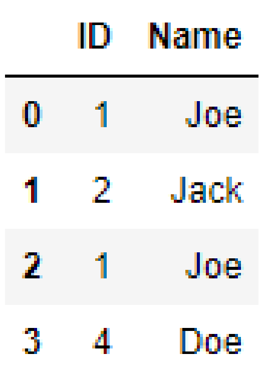
ID	NAME
1	Joe
2	Jack
1	Joe
4	Doe



```python
import pandas as pd
```

```python
df1 = {
    'ID': [1, 2],
    'Name': ['Joe', 'Jack']
}

df1 = pd.DataFrame(df1)

df2 = {
    'ID': [1, 4],
    'Name': ['Joe', 'Doe']
}

df2 = pd.DataFrame(df2)
```

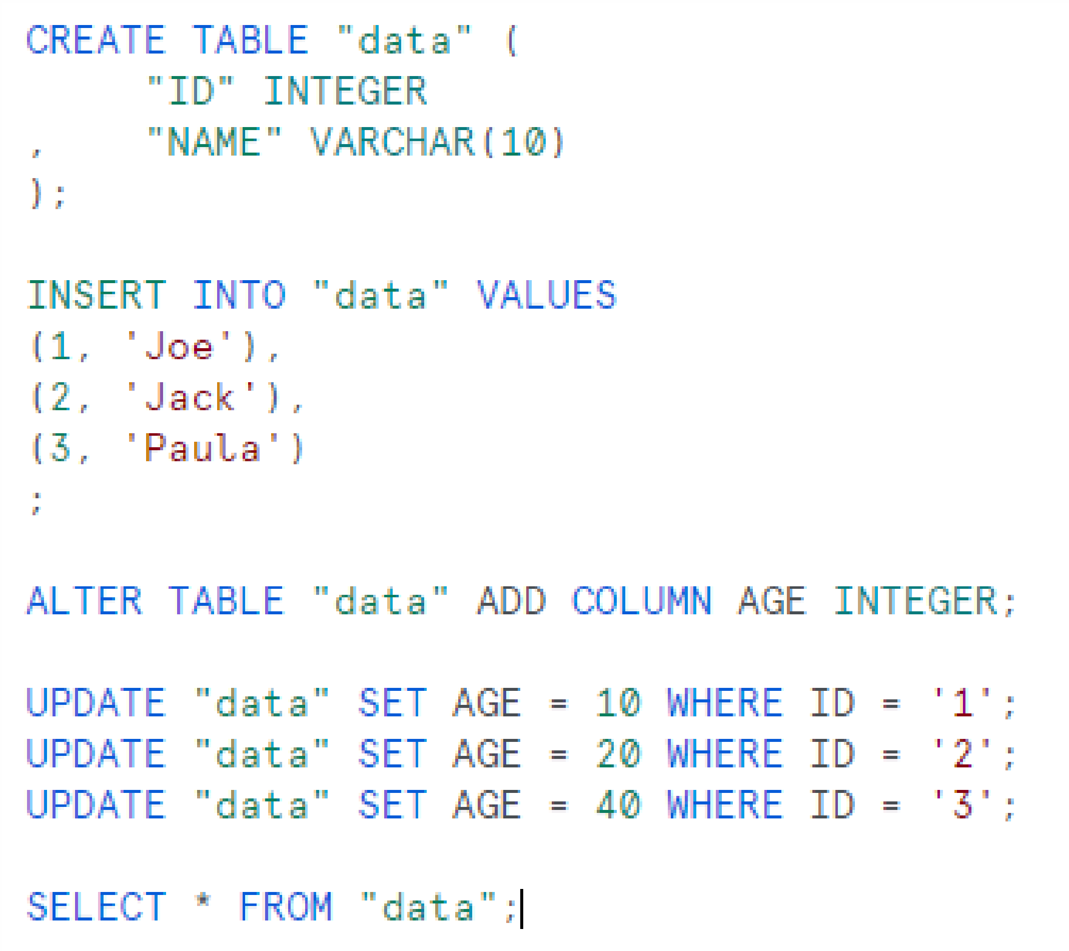
```python
df = pd.concat([df1, df2], ignore_index=True)

# ignore_index=True will reindex the dataframe

df
```



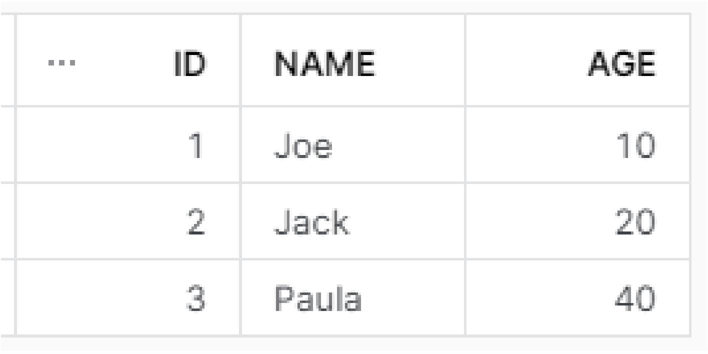
	ID	Name
0	1	Joe
1	2	Jack
2	1	Joe
3	4	Doe

in Uzwal Goud Vaddeboina

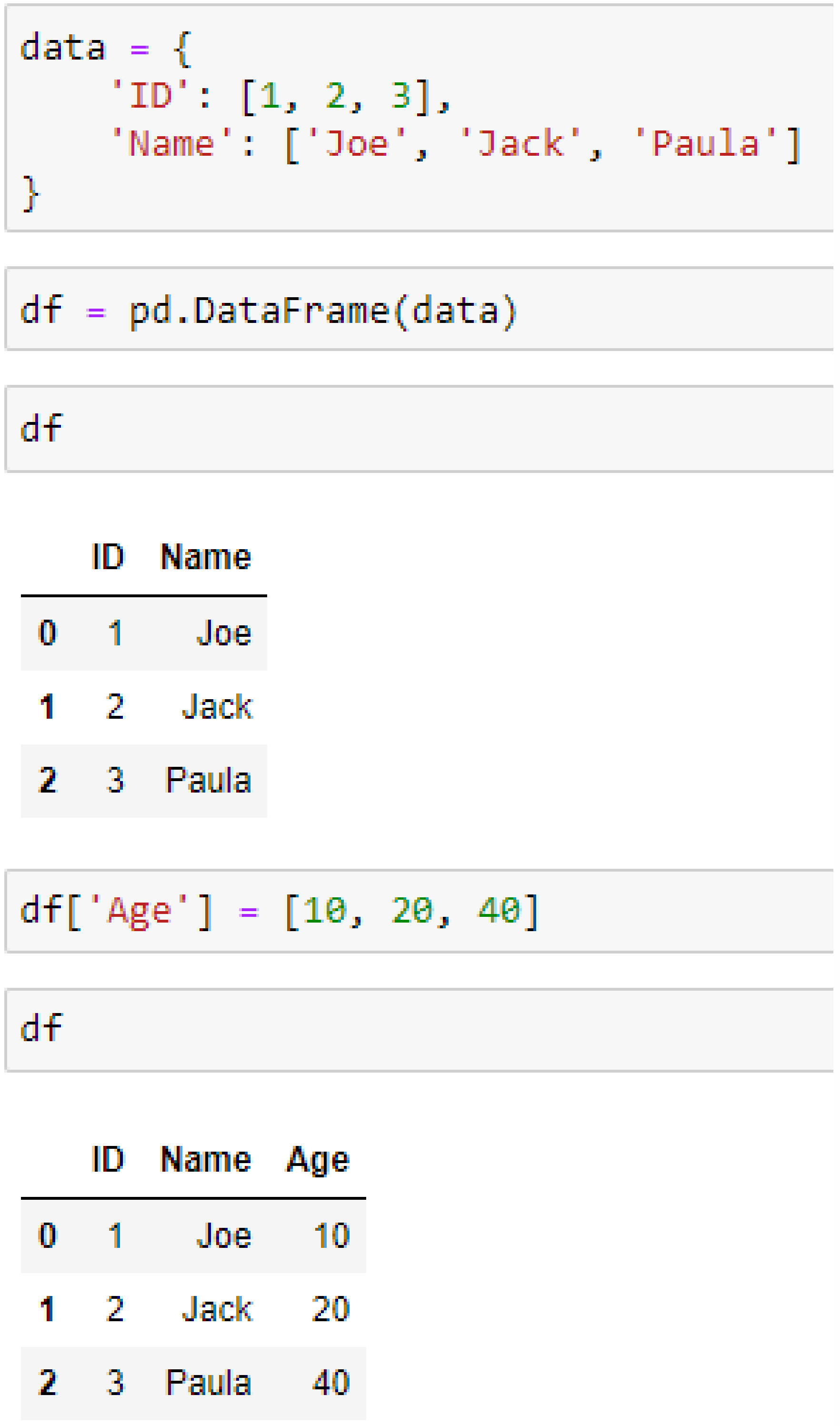
ADD SINGLE COLUMN



```sql
CREATE TABLE "data" (
     "ID" INTEGER
,    "NAME" VARCHAR(10)
);

INSERT INTO "data" VALUES
(1, 'Joe'),
(2, 'Jack'),
(3, 'Paula')
;

ALTER TABLE "data" ADD COLUMN AGE INTEGER;

UPDATE "data" SET AGE = 10 WHERE ID = '1';
UPDATE "data" SET AGE = 20 WHERE ID = '2';
UPDATE "data" SET AGE = 40 WHERE ID = '3';

SELECT * FROM "data";
```



···	ID	NAME	AGE
	1	Joe	10
	2	Jack	20
	3	Paula	40



```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Jack', 'Paula']
}
```

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

```python
df
```

	ID	Name
0	1	Joe
1	2	Jack
2	3	Paula

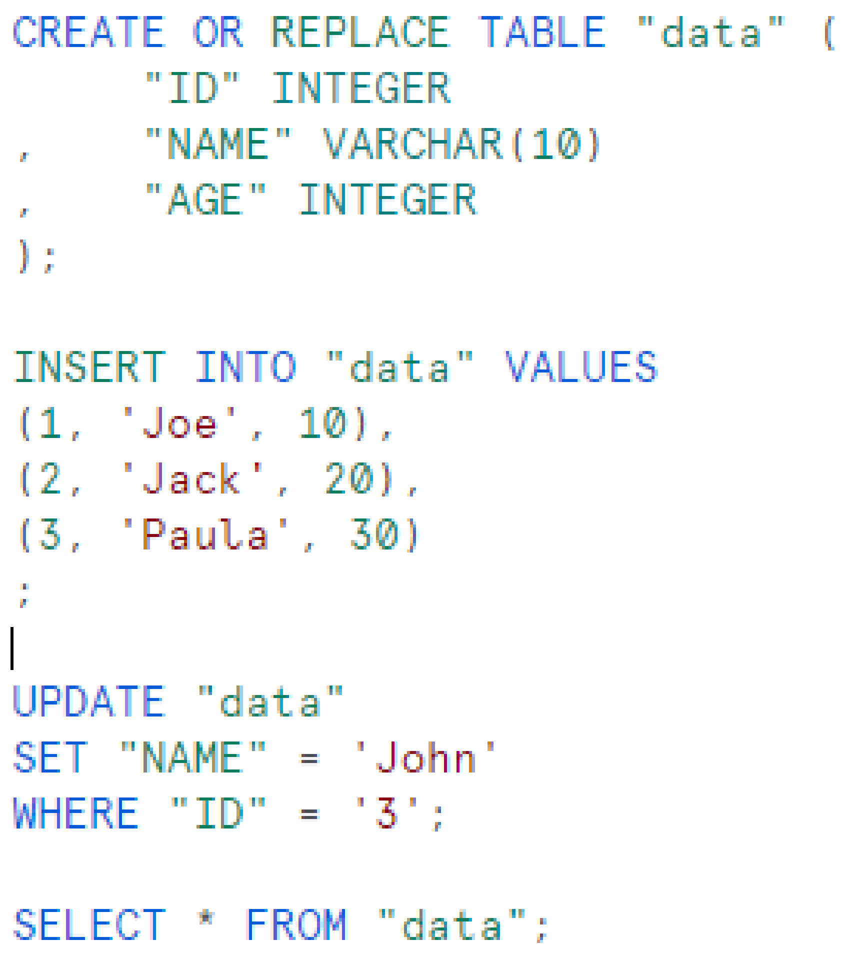
```python
df['Age'] = [10, 20, 40]
```

```python
df
```

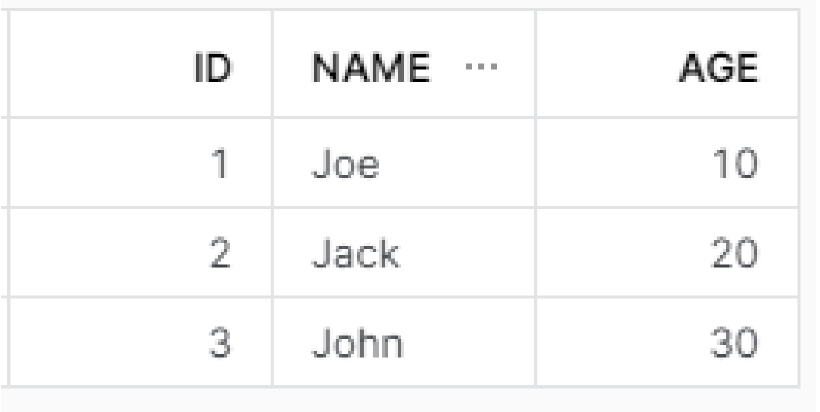
	ID	Name	Age
0	1	Joe	10
1	2	Jack	20
2	3	Paula	40

Uzwal Goud Vaddeboina

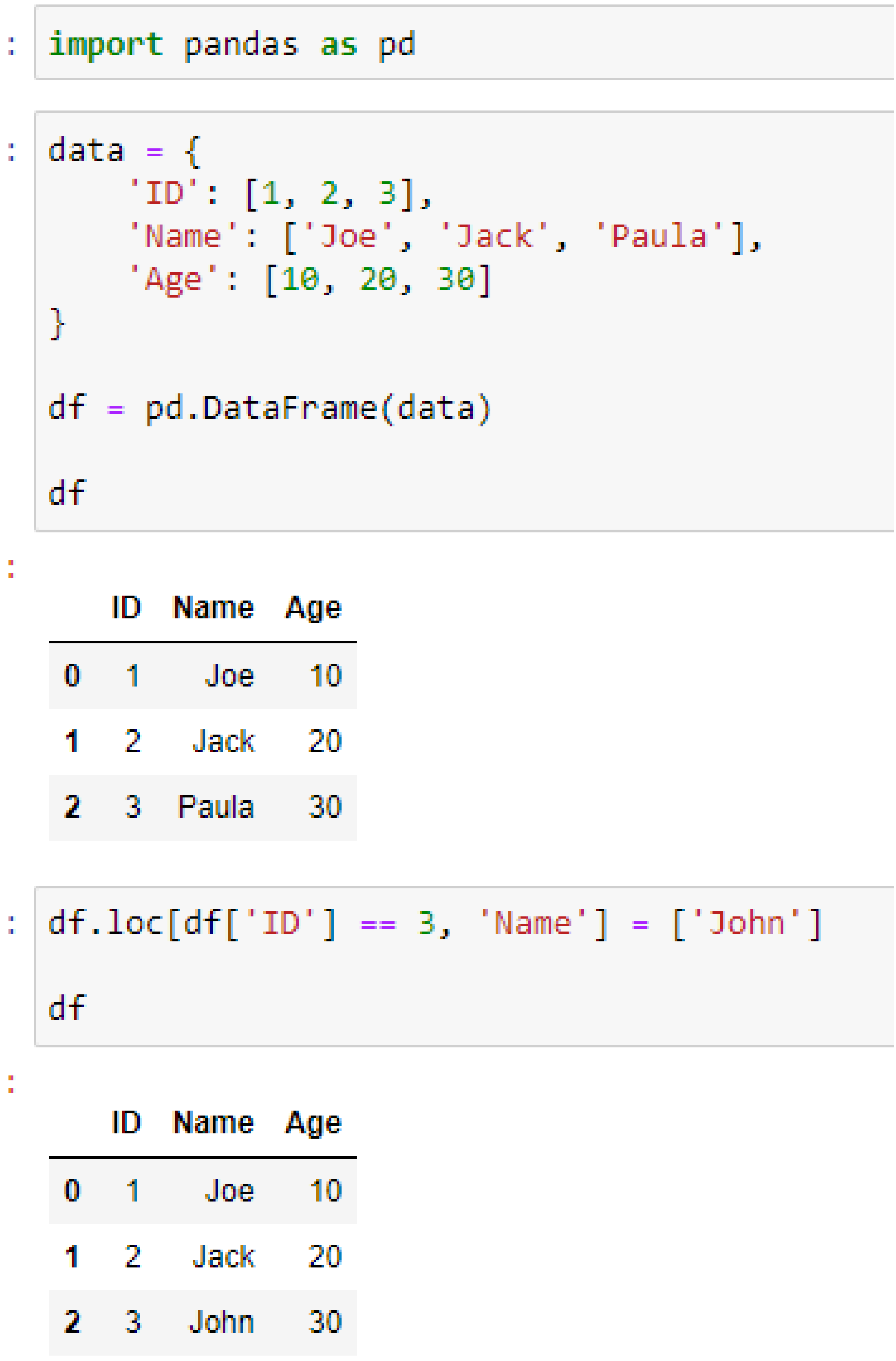
UPDATE SINGLE COLUMN SINGLE ROW



```sql
CREATE OR REPLACE TABLE "data" (
      "ID" INTEGER
,     "NAME" VARCHAR(10)
,     "AGE" INTEGER
);

INSERT INTO "data" VALUES
(1, 'Joe', 10),
(2, 'Jack', 20),
(3, 'Paula', 30)
;

UPDATE "data"
SET "NAME" = 'John'
WHERE "ID" = '3';

SELECT * FROM "data";
```



ID	NAME ···	AGE
1	Joe	10
2	Jack	20
3	John	30



```python
import pandas as pd
```

```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Jack', 'Paula'],
    'Age': [10, 20, 30]
}

df = pd.DataFrame(data)

df
```

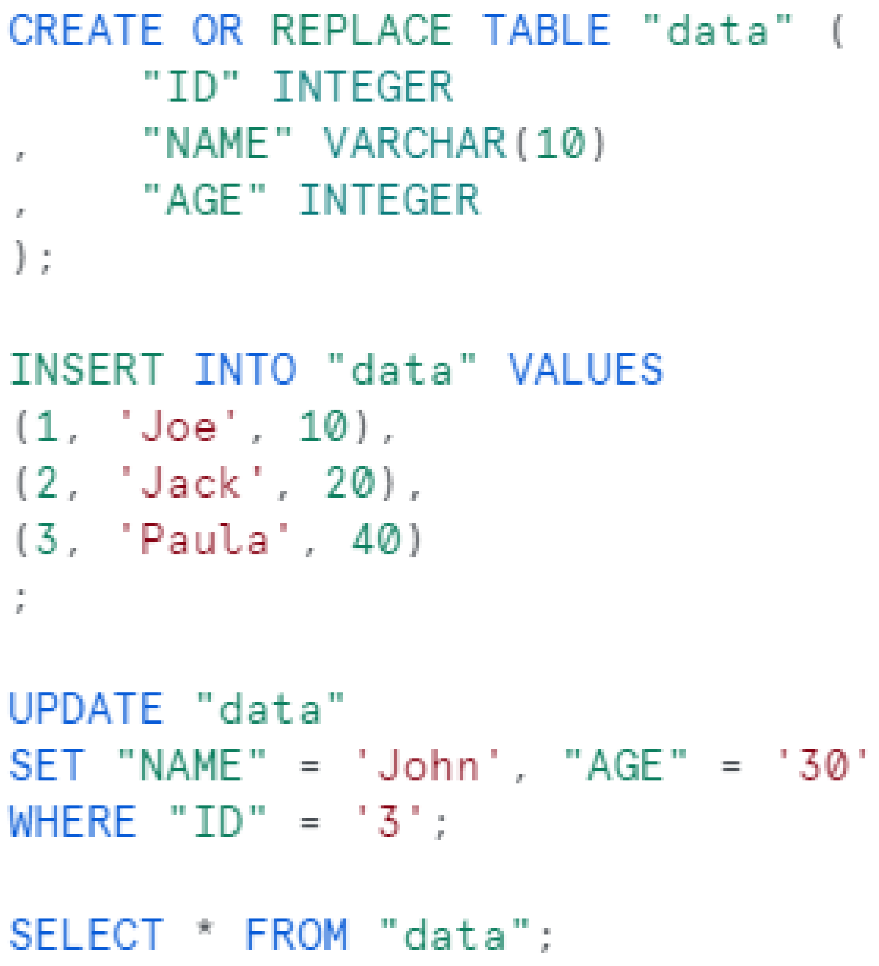
	ID	Name	Age
0	1	Joe	10
1	2	Jack	20
2	3	Paula	30

```python
df.loc[df['ID'] == 3, 'Name'] = ['John']

df
```

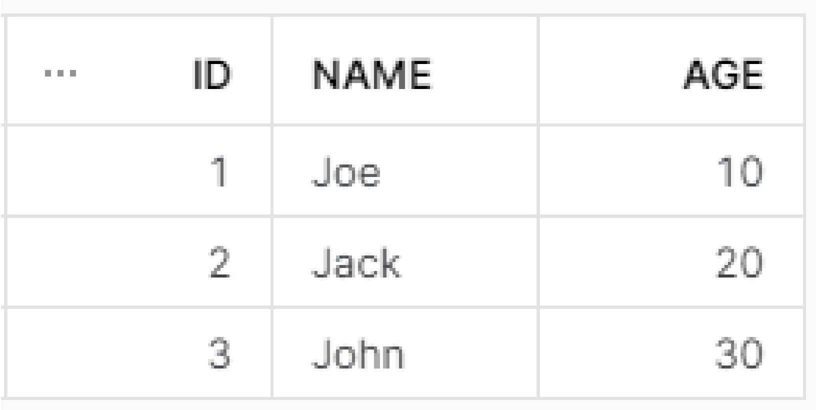
	ID	Name	Age
0	1	Joe	10
1	2	Jack	20
2	3	John	30

Uzwal Goud Vaddeboina

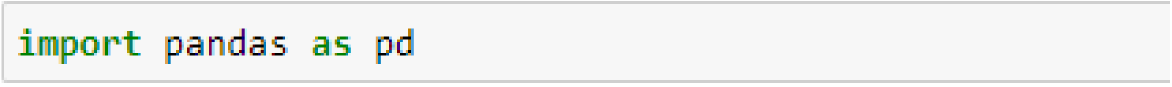
UPDATE MULTIPLE COLUMNS SINGLE ROW



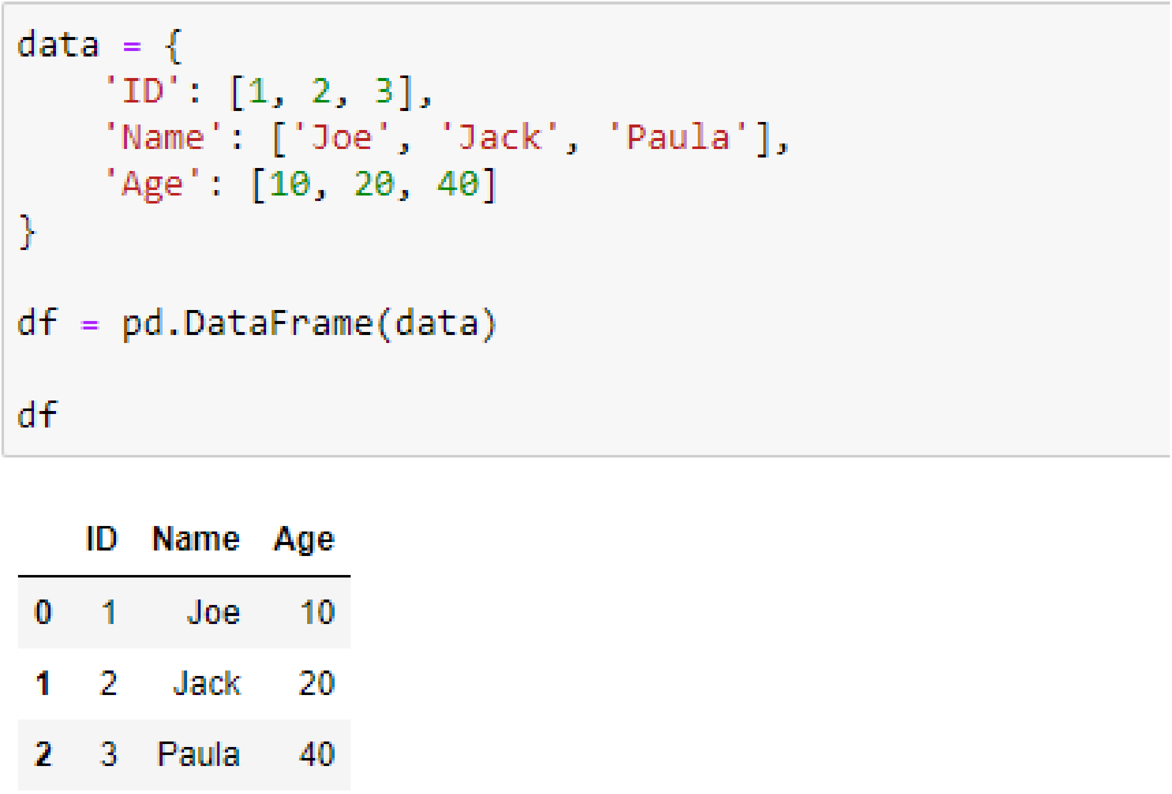
```sql
CREATE OR REPLACE TABLE "data" (
     "ID" INTEGER
,    "NAME" VARCHAR(10)
,    "AGE" INTEGER
);

INSERT INTO "data" VALUES
(1, 'Joe', 10),
(2, 'Jack', 20),
(3, 'Paula', 40)
;

UPDATE "data"
SET "NAME" = 'John', "AGE" = '30'
WHERE "ID" = '3';

SELECT * FROM "data";
```



...	ID	NAME	AGE
	1	Joe	10
	2	Jack	20
	3	John	30

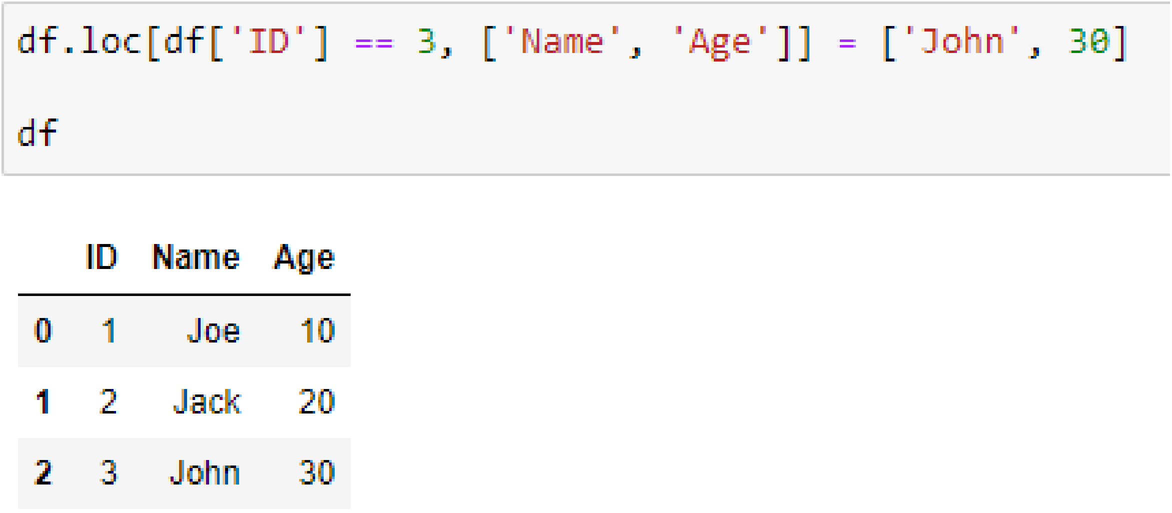


```python
import pandas as pd
```



```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Jack', 'Paula'],
    'Age': [10, 20, 40]
}

df = pd.DataFrame(data)

df
```

	ID	Name	Age
0	1	Joe	10
1	2	Jack	20
2	3	Paula	40

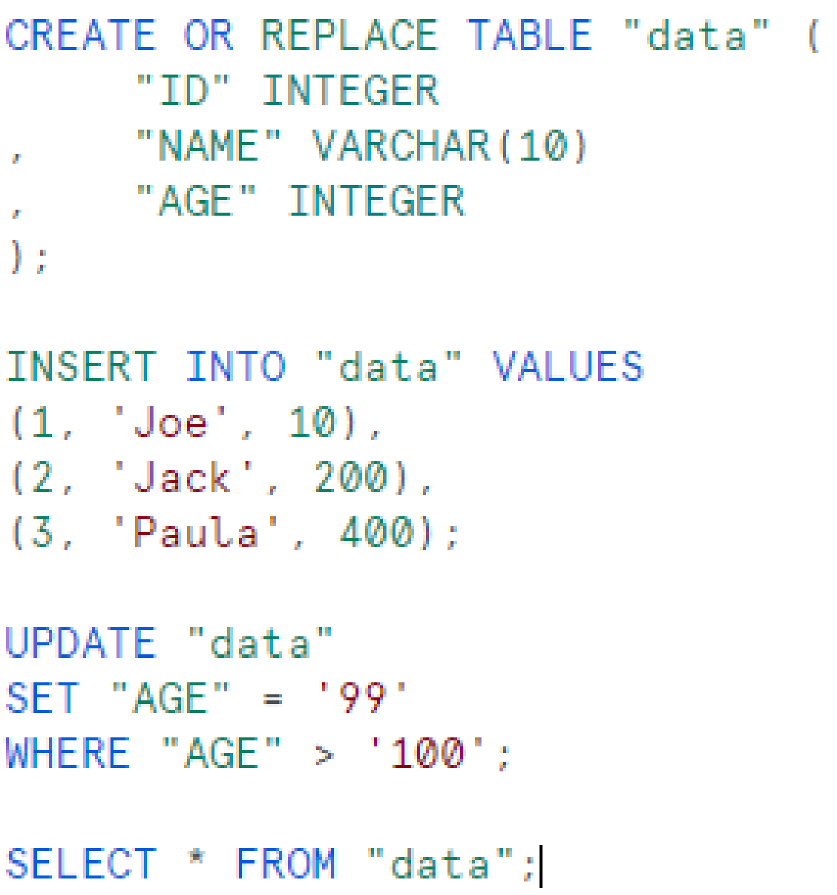


```python
df.loc[df['ID'] == 3, ['Name', 'Age']] = ['John', 30]

df
```

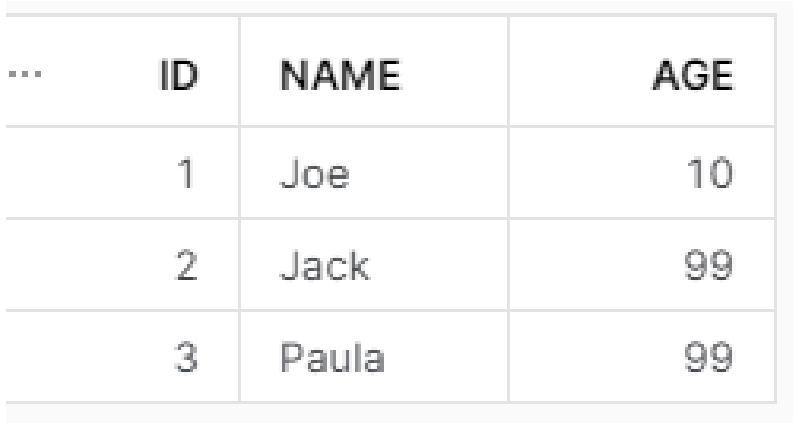
	ID	Name	Age
0	1	Joe	10
1	2	Jack	20
2	3	John	30

in Uzwal Goud Vaddeboina

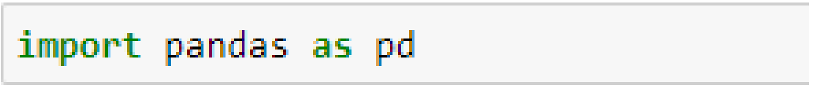
UPDATE SINGLE COLUMN MULTIPLE ROWS



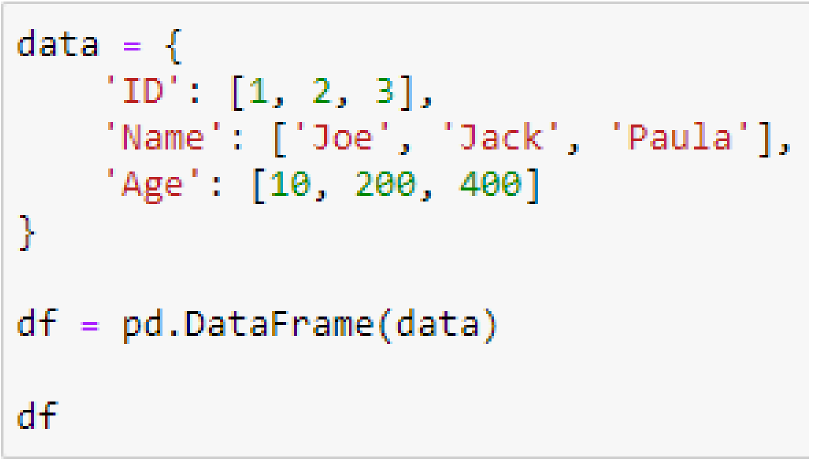
```sql
CREATE OR REPLACE TABLE "data" (
     "ID" INTEGER
,    "NAME" VARCHAR(10)
,    "AGE" INTEGER
);

INSERT INTO "data" VALUES
(1, 'Joe', 10),
(2, 'Jack', 200),
(3, 'Paula', 400);

UPDATE "data"
SET "AGE" = '99'
WHERE "AGE" > '100';

SELECT * FROM "data";
```



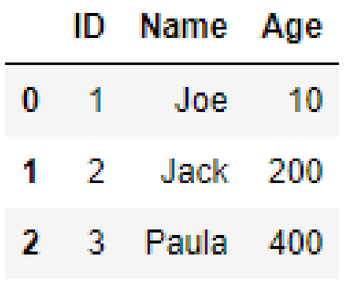
...	ID	NAME	AGE
	1	Joe	10
	2	Jack	99
	3	Paula	99



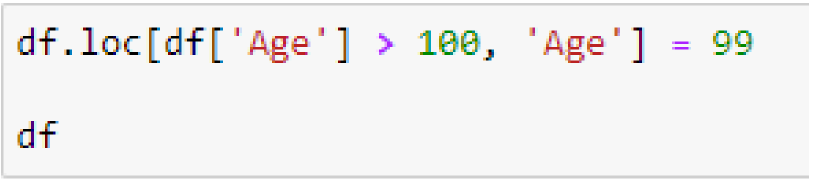
```python
import pandas as pd
```



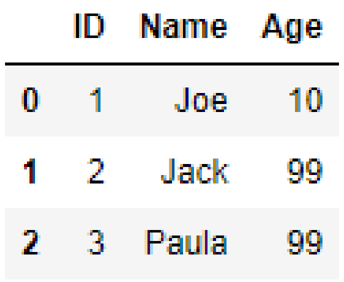
```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Jack', 'Paula'],
    'Age': [10, 200, 400]
}

df = pd.DataFrame(data)

df
```



	ID	Name	Age
0	1	Joe	10
1	2	Jack	200
2	3	Paula	400



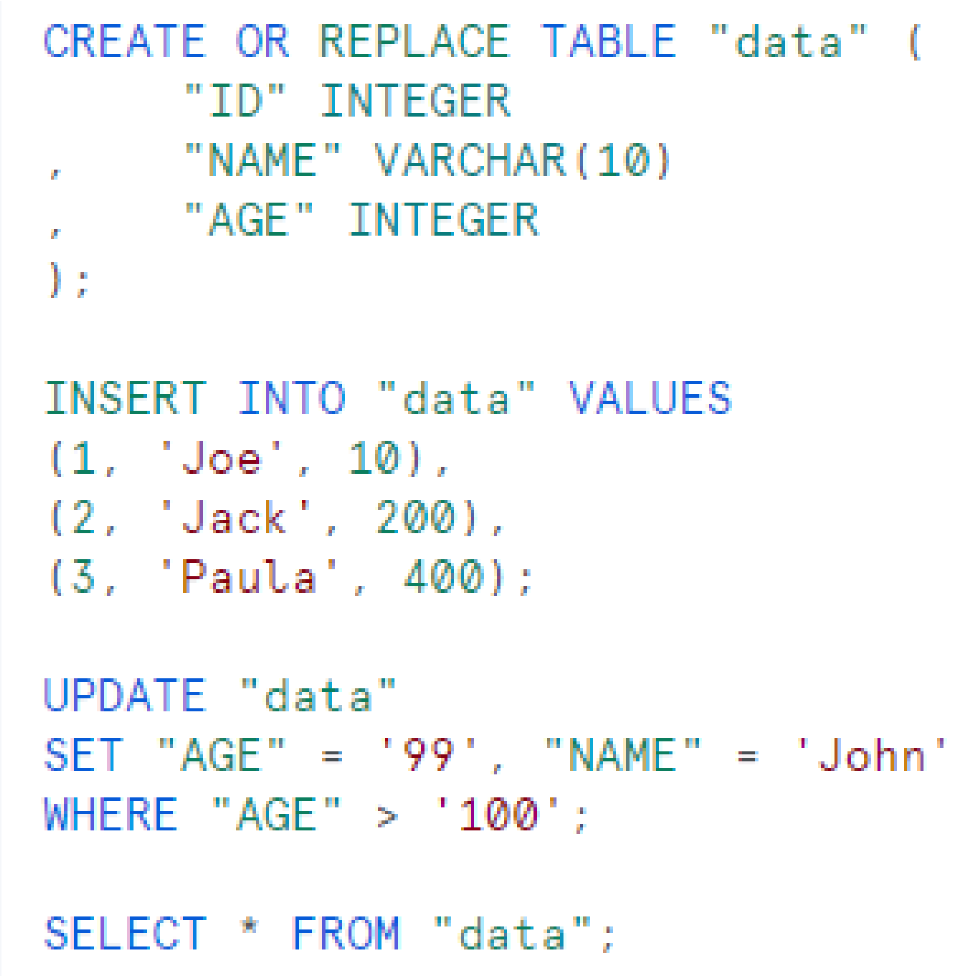
```python
df.loc[df['Age'] > 100, 'Age'] = 99

df
```



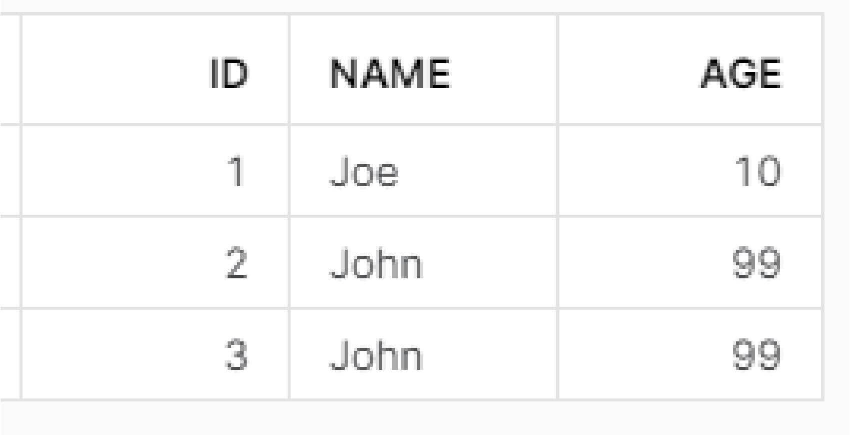
	ID	Name	Age
0	1	Joe	10
1	2	Jack	99
2	3	Paula	99

Uzwal Goud Vaddeboina

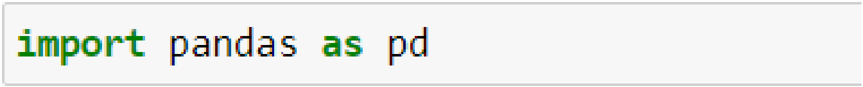
UPDATE MULTIPLE COLUMNS MULTIPLE ROWS



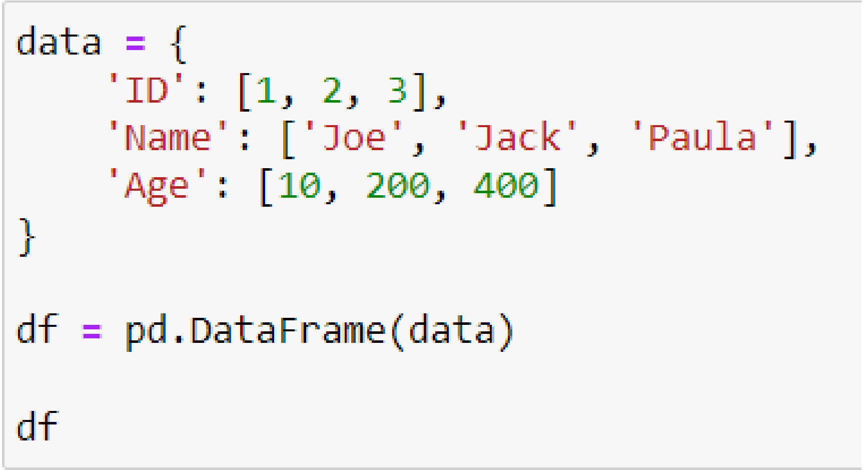
```sql
CREATE OR REPLACE TABLE "data" (
     "ID" INTEGER
,    "NAME" VARCHAR(10)
,    "AGE" INTEGER
);

INSERT INTO "data" VALUES
(1, 'Joe', 10),
(2, 'Jack', 200),
(3, 'Paula', 400);

UPDATE "data"
SET "AGE" = '99', "NAME" = 'John'
WHERE "AGE" > '100';

SELECT * FROM "data";
```



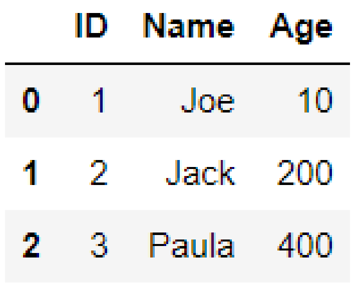
ID	NAME	AGE
1	Joe	10
2	John	99
3	John	99



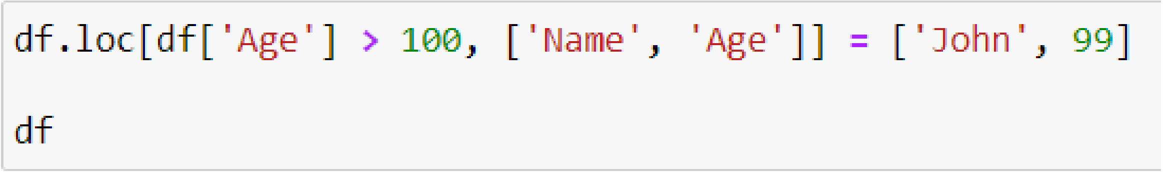
```python
import pandas as pd
```



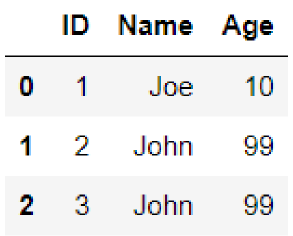
```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Jack', 'Paula'],
    'Age': [10, 200, 400]
}

df = pd.DataFrame(data)

df
```



	ID	Name	Age
0	1	Joe	10
1	2	Jack	200
2	3	Paula	400



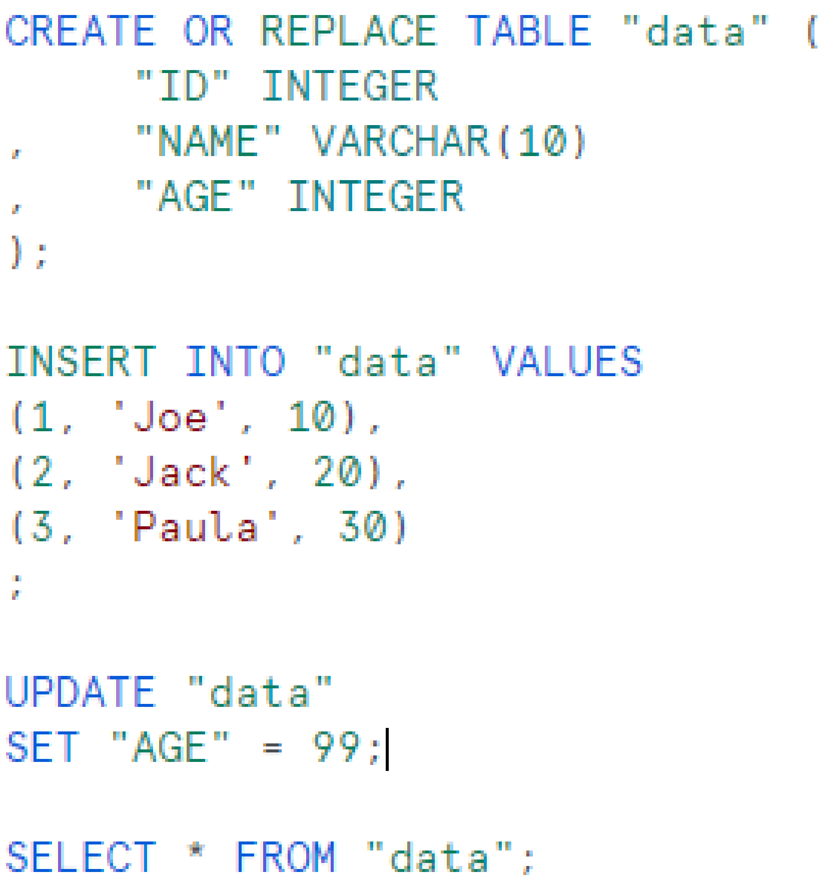
```python
df.loc[df['Age'] > 100, ['Name', 'Age']] = ['John', 99]

df
```



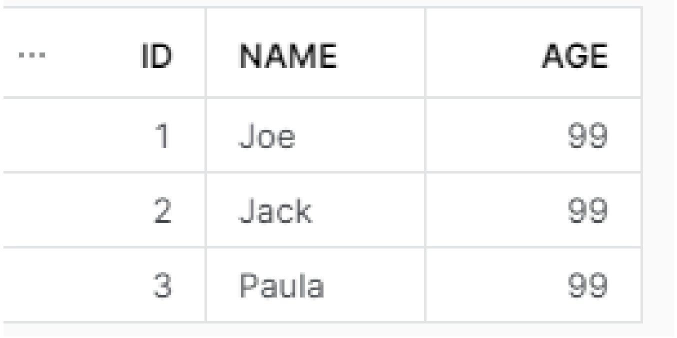
	ID	Name	Age
0	1	Joe	10
1	2	John	99
2	3	John	99

in Uzwal Goud Vaddeboina

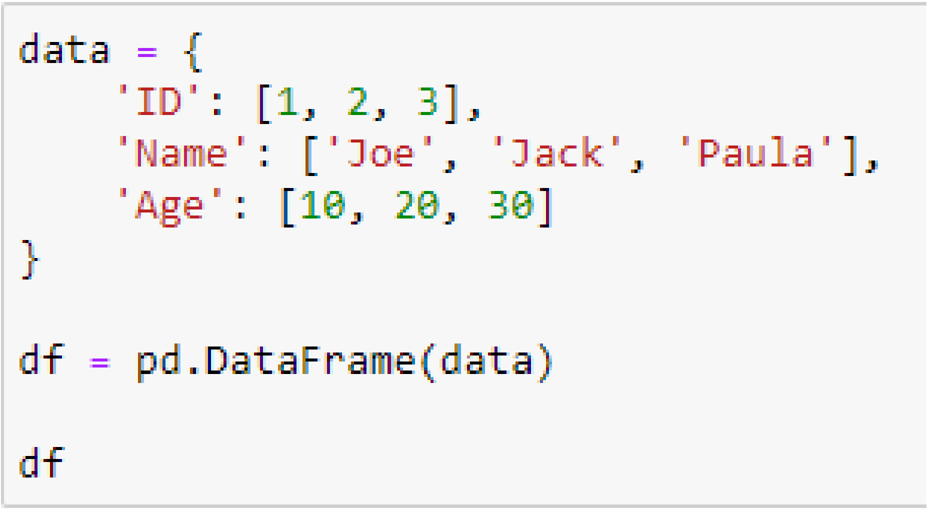
UPDATE SINGLE COLUMN ALL ROWS



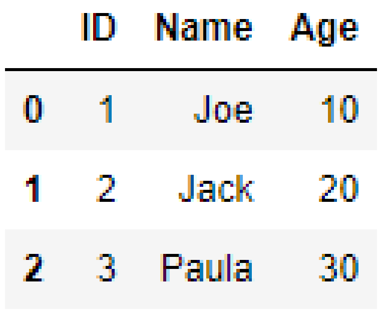
```sql
CREATE OR REPLACE TABLE "data" (
     "ID" INTEGER
,    "NAME" VARCHAR(10)
,    "AGE" INTEGER
);

INSERT INTO "data" VALUES
(1, 'Joe', 10),
(2, 'Jack', 20),
(3, 'Paula', 30)
;

UPDATE "data"
SET "AGE" = 99;

SELECT * FROM "data";
```



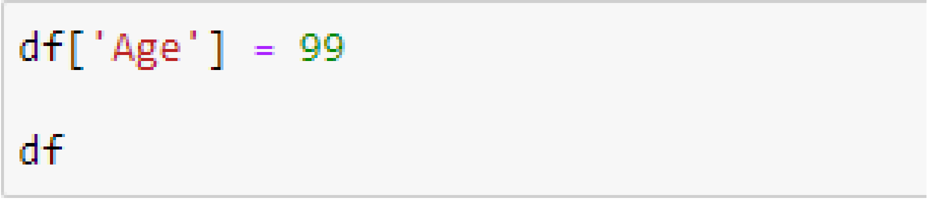
···	ID	NAME	AGE
	1	Joe	99
	2	Jack	99
	3	Paula	99



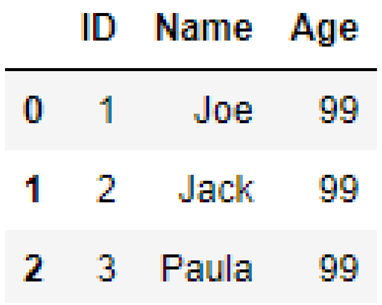
```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Jack', 'Paula'],
    'Age': [10, 20, 30]
}

df = pd.DataFrame(data)

df
```



	ID	Name	Age
0	1	Joe	10
1	2	Jack	20
2	3	Paula	30



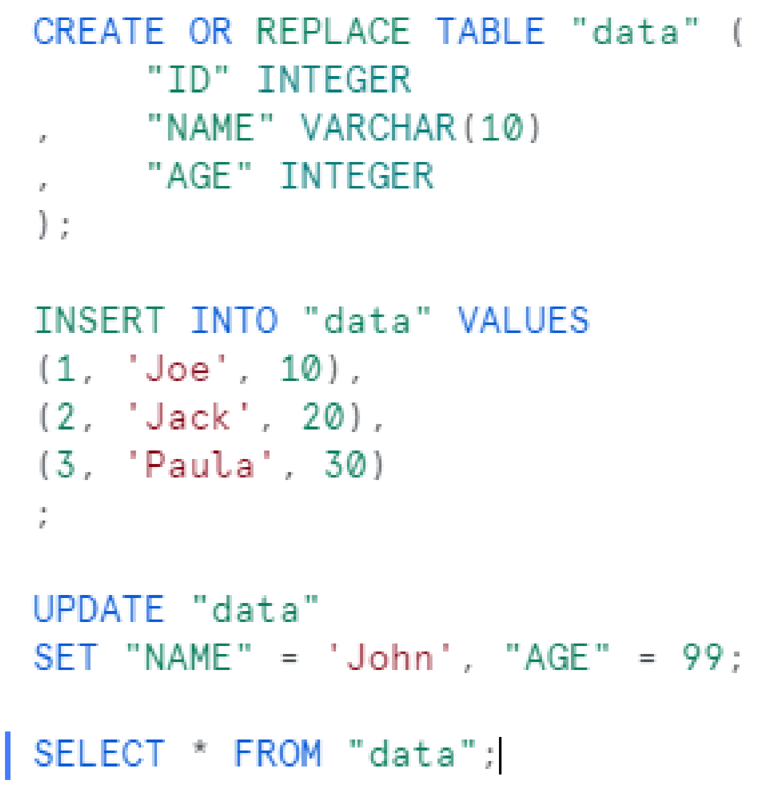
```python
df['Age'] = 99

df
```



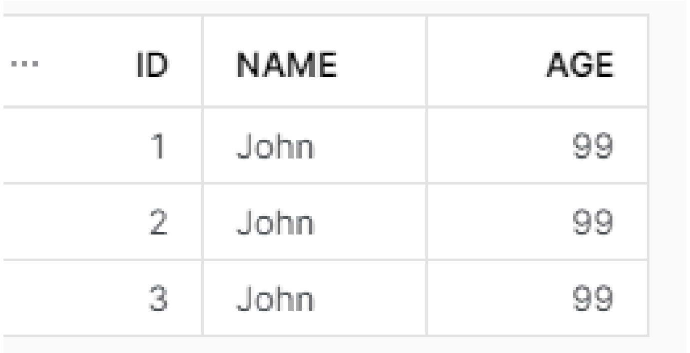
	ID	Name	Age
0	1	Joe	99
1	2	Jack	99
2	3	Paula	99

in Uzwal Goud Vaddeboina

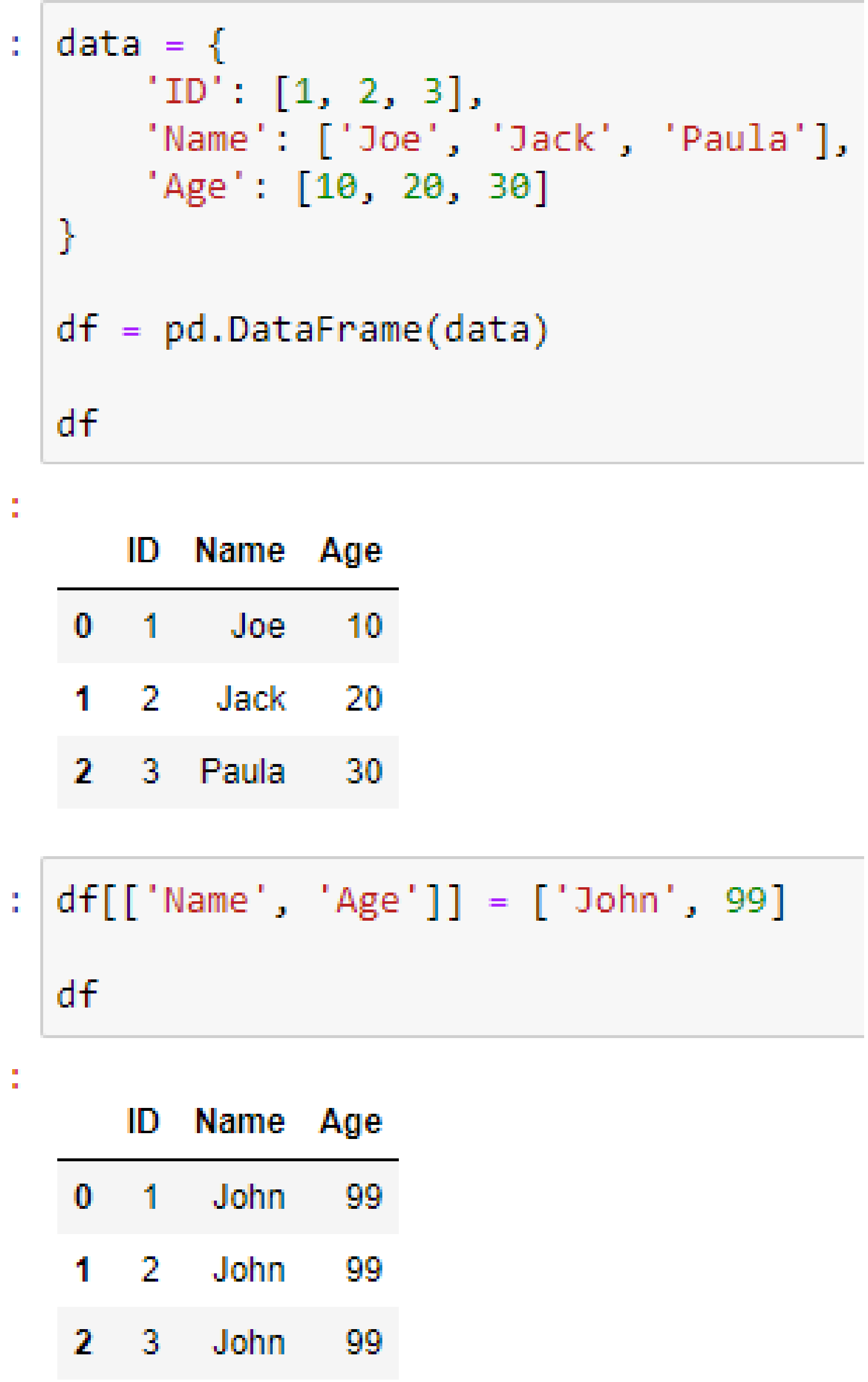
UPDATE MULTIPLE COLUMNS ALL ROWS



```sql
CREATE OR REPLACE TABLE "data" (
      "ID" INTEGER
,     "NAME" VARCHAR(10)
,     "AGE" INTEGER
);

INSERT INTO "data" VALUES
(1, 'Joe', 10),
(2, 'Jack', 20),
(3, 'Paula', 30)
;

UPDATE "data"
SET "NAME" = 'John', "AGE" = 99;

SELECT * FROM "data";
```



··· ID	NAME	AGE
1	John	99
2	John	99
3	John	99



```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Jack', 'Paula'],
    'Age': [10, 20, 30]
}

df = pd.DataFrame(data)

df
```

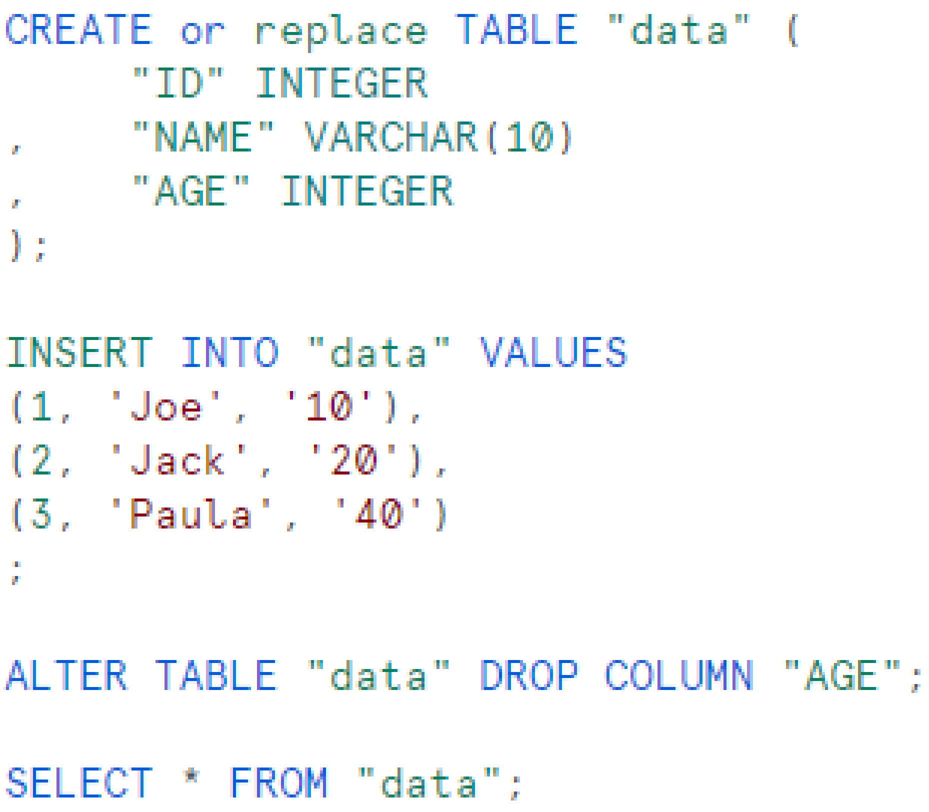
	ID	Name	Age
0	1	Joe	10
1	2	Jack	20
2	3	Paula	30

```python
df[['Name', 'Age']] = ['John', 99]

df
```

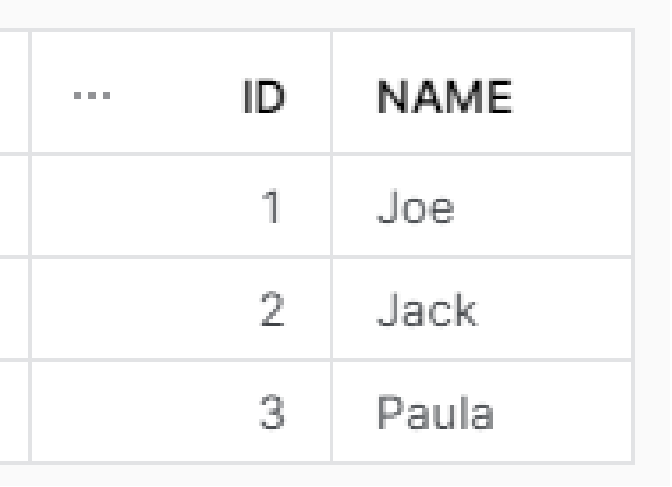
	ID	Name	Age
0	1	John	99
1	2	John	99
2	3	John	99

Uzwal Goud Vaddeboina

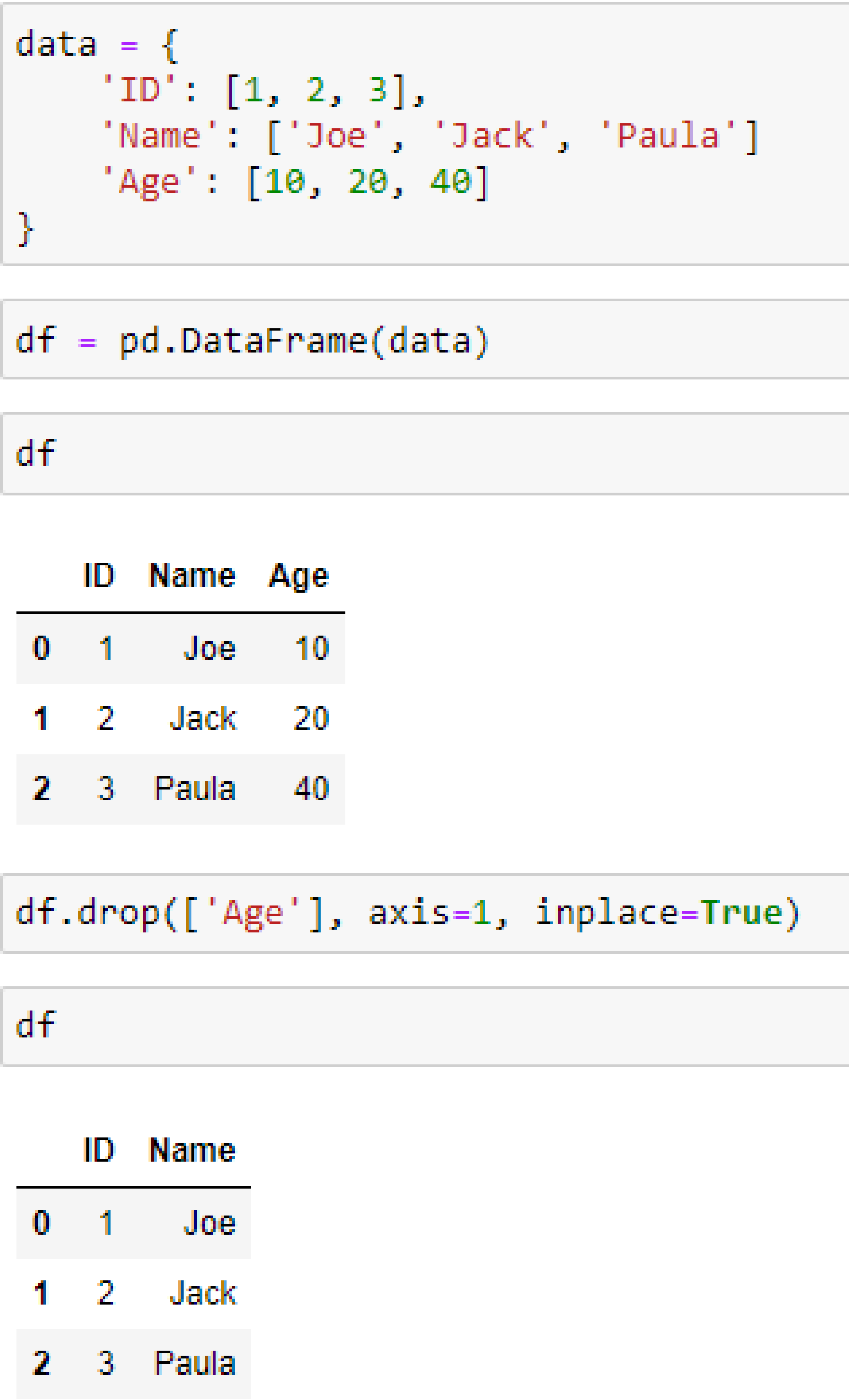
DROP SINGLE COLUMN



```sql
CREATE or replace TABLE "data" (
      "ID" INTEGER
,     "NAME" VARCHAR(10)
,     "AGE" INTEGER
);

INSERT INTO "data" VALUES
(1, 'Joe', '10'),
(2, 'Jack', '20'),
(3, 'Paula', '40')
;

ALTER TABLE "data" DROP COLUMN "AGE";

SELECT * FROM "data";
```



...	ID	NAME
	1	Joe
	2	Jack
	3	Paula



```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Jack', 'Paula'],
    'Age': [10, 20, 40]
}
```

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

```python
df
```

```
   ID  Name  Age
0   1   Joe   10
1   2  Jack   20
2   3 Paula   40
```

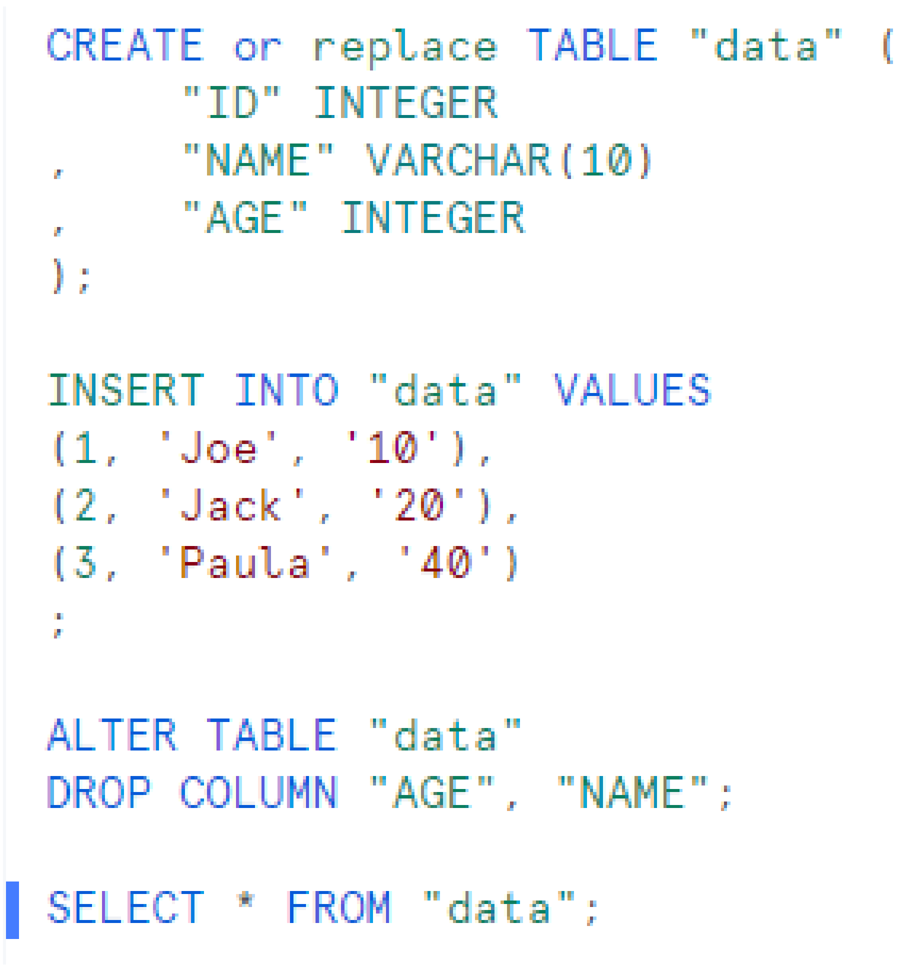
```python
df.drop(['Age'], axis=1, inplace=True)
```

```python
df
```

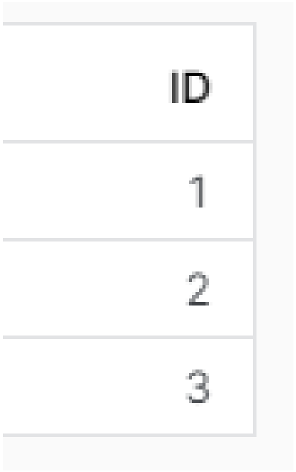
```
   ID  Name
0   1   Joe
1   2  Jack
2   3 Paula
```

Uzwal Goud Vaddeboina

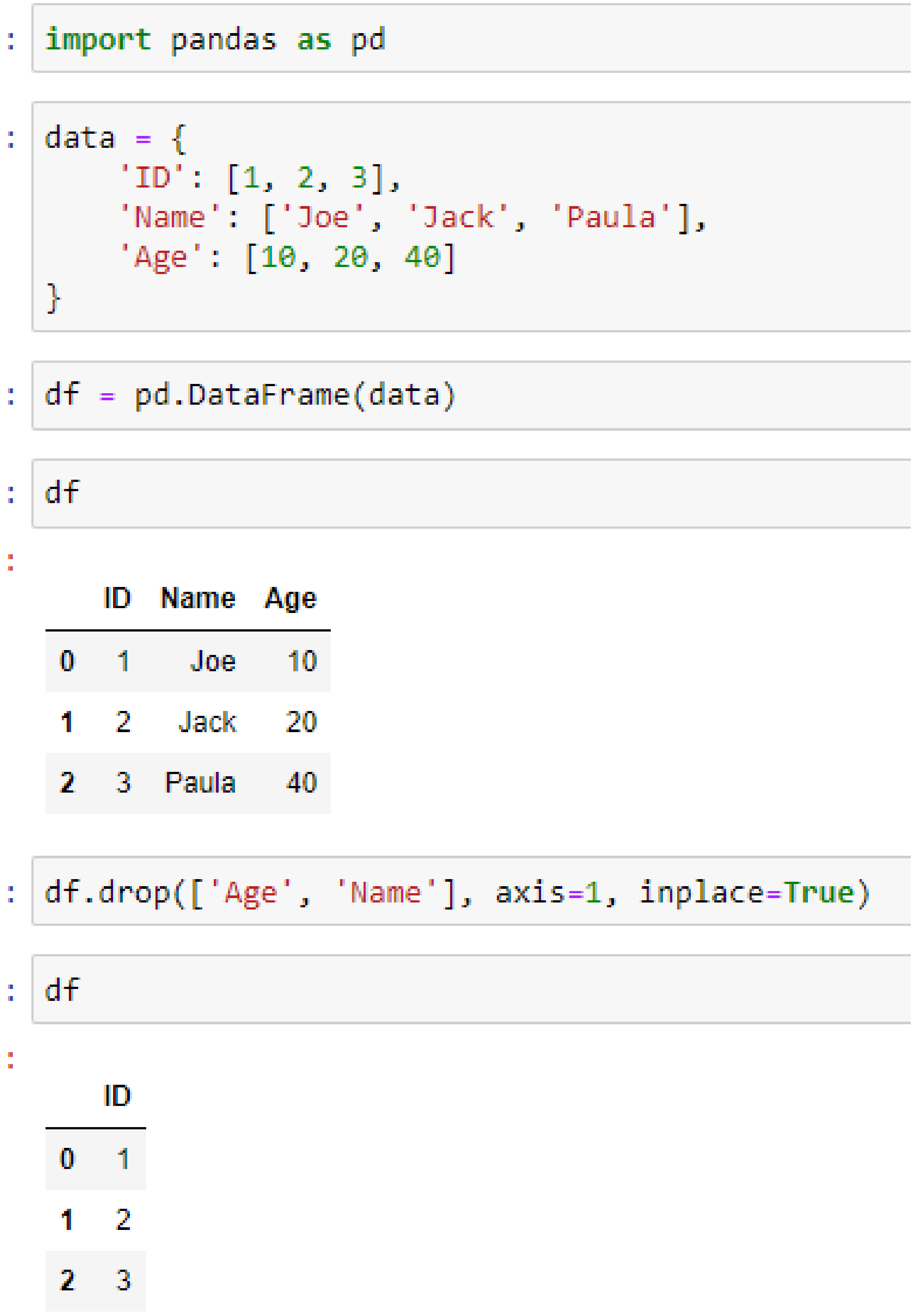
DROP MULTIPLE COLUMNS



```sql
CREATE or replace TABLE "data" (
     "ID" INTEGER
,    "NAME" VARCHAR(10)
,    "AGE" INTEGER
);

INSERT INTO "data" VALUES
(1, 'Joe', '10'),
(2, 'Jack', '20'),
(3, 'Paula', '40')
;

ALTER TABLE "data"
DROP COLUMN "AGE", "NAME";

SELECT * FROM "data";
```



ID
1
2
3



```python
import pandas as pd
```

```python
data = {
    'ID': [1, 2, 3],
    'Name': ['Joe', 'Jack', 'Paula'],
    'Age': [10, 20, 40]
}
```

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

```python
df
```

	ID	Name	Age
0	1	Joe	10
1	2	Jack	20
2	3	Paula	40

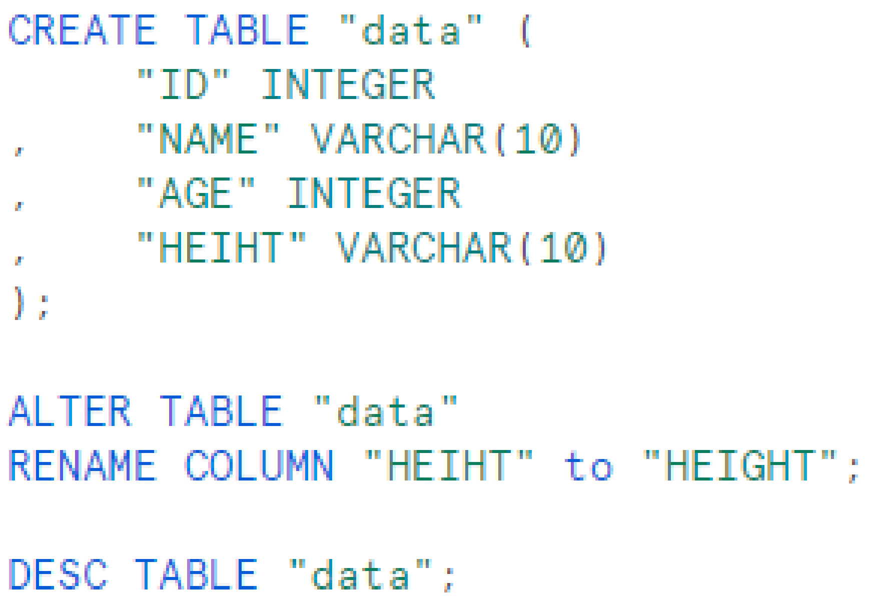
```python
df.drop(['Age', 'Name'], axis=1, inplace=True)
```

```python
df
```

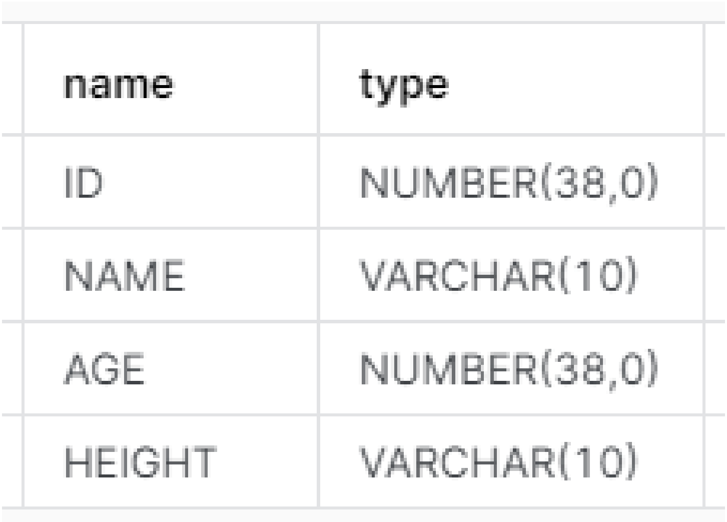
	ID
0	1
1	2
2	3

Uzwal Goud Vaddeboina

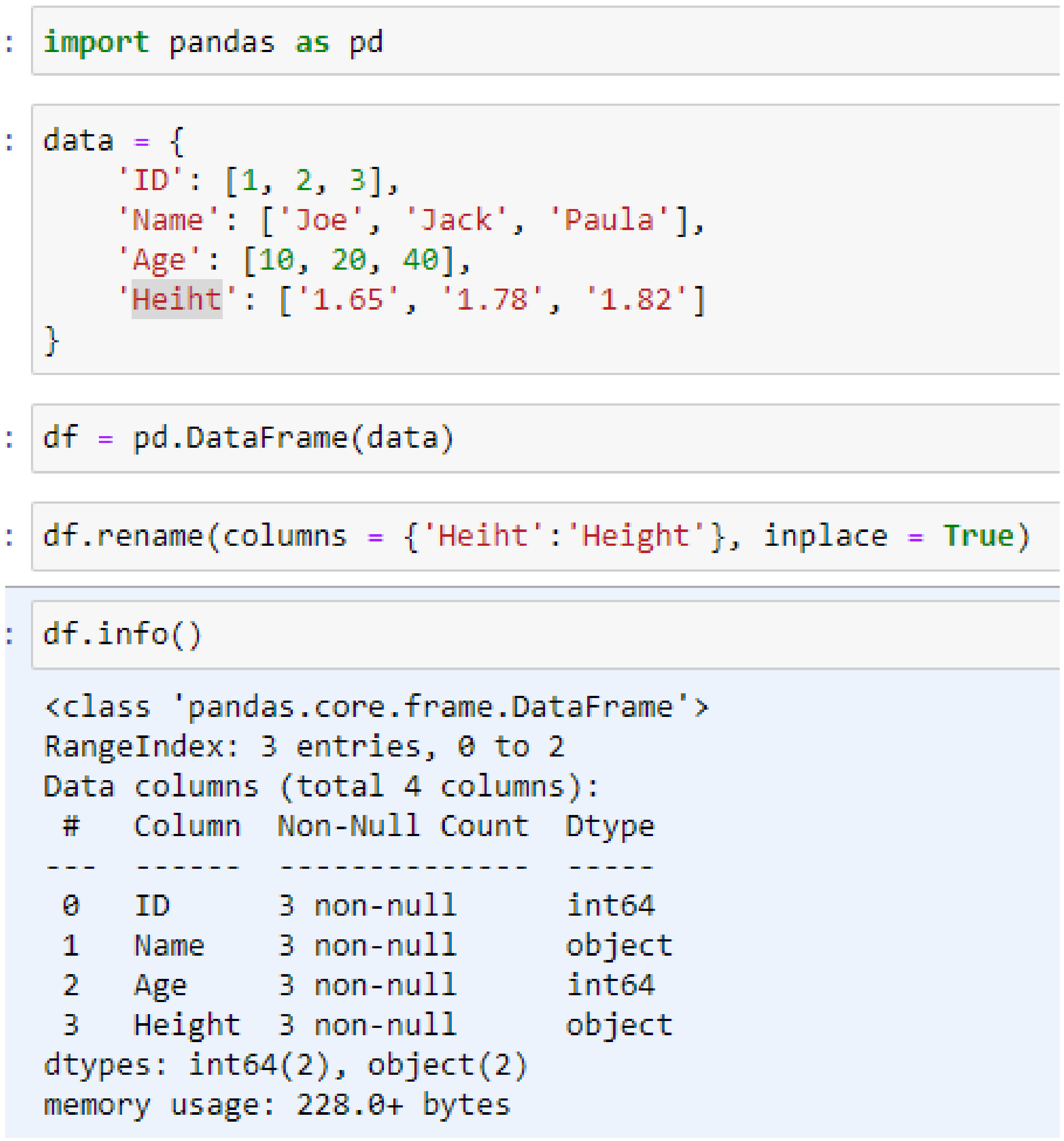
RENAME SINGLE COLUMN



```sql
CREATE TABLE "data" (
     "ID" INTEGER
,    "NAME" VARCHAR(10)
,    "AGE" INTEGER
,    "HEIHT" VARCHAR(10)
);

ALTER TABLE "data"
RENAME COLUMN "HEIHT" to "HEIGHT";

DESC TABLE "data";
```



name	type
ID	NUMBER(38,0)
NAME	VARCHAR(10)
AGE	NUMBER(38,0)
HEIGHT	VARCHAR(10)

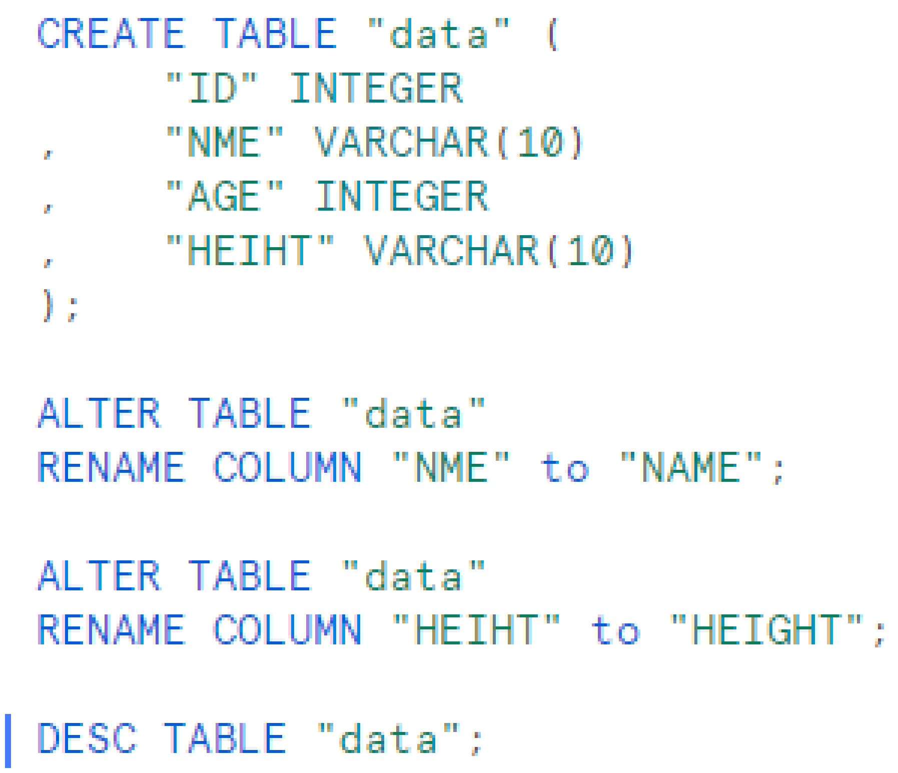


```python
: import pandas as pd

: data = {
      'ID': [1, 2, 3],
      'Name': ['Joe', 'Jack', 'Paula'],
      'Age': [10, 20, 40],
      'Heiht': ['1.65', '1.78', '1.82']
  }

: df = pd.DataFrame(data)

: df.rename(columns = {'Heiht':'Height'}, inplace = True)

: df.info()
```

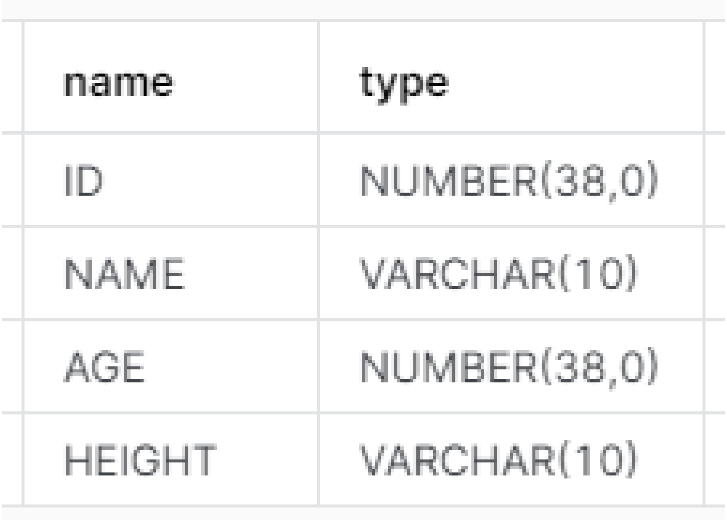
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   ID      3 non-null      int64 
 1   Name    3 non-null      object
 2   Age     3 non-null      int64 
 3   Height  3 non-null      object
dtypes: int64(2), object(2)
memory usage: 228.0+ bytes
```

in Uzwal Goud Vaddeboina

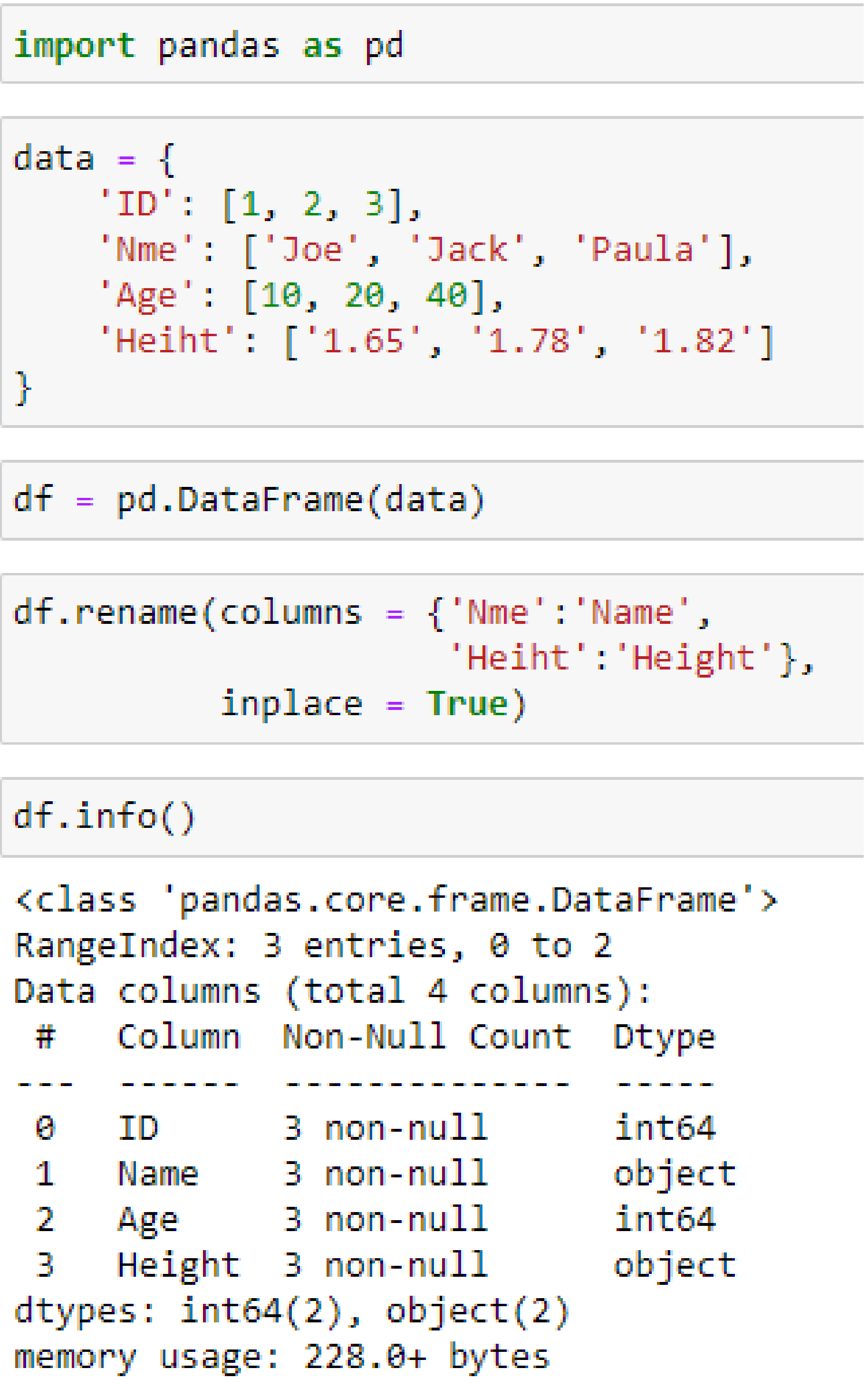
RENAME MULTIPLE COLUMNS



```
CREATE TABLE "data" (
      "ID" INTEGER
,     "NME" VARCHAR(10)
,     "AGE" INTEGER
,     "HEIHT" VARCHAR(10)
);

ALTER TABLE "data"
RENAME COLUMN "NME" to "NAME";

ALTER TABLE "data"
RENAME COLUMN "HEIHT" to "HEIGHT";

DESC TABLE "data";
```



name	type
ID	NUMBER(38,0)
NAME	VARCHAR(10)
AGE	NUMBER(38,0)
HEIGHT	VARCHAR(10)



```
import pandas as pd
```

```
data = {
    'ID': [1, 2, 3],
    'Nme': ['Joe', 'Jack', 'Paula'],
    'Age': [10, 20, 40],
    'Heiht': ['1.65', '1.78', '1.82']
}
```

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

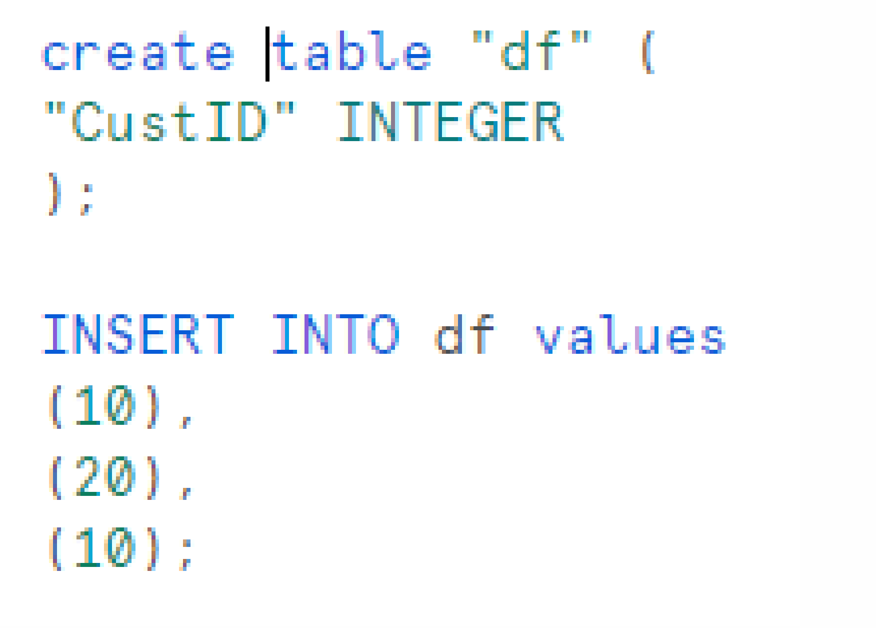
```
df.rename(columns = {'Nme':'Name',
                     'Heiht':'Height'},
          inplace = True)
```

```
df.info()
```

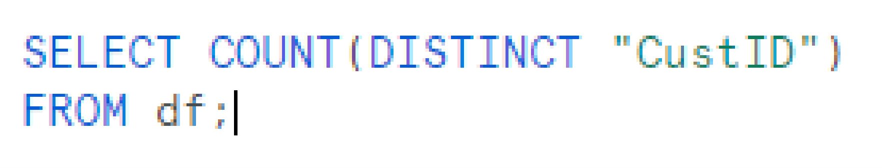
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   ID      3 non-null      int64 
 1   Name    3 non-null      object
 2   Age     3 non-null      int64 
 3   Height  3 non-null      object
dtypes: int64(2), object(2)
memory usage: 228.0+ bytes
```

Uzwal Goud Vaddeboina

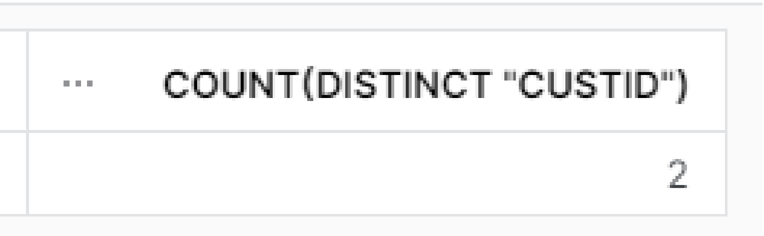
COUNT OF DISTINCT VALUES



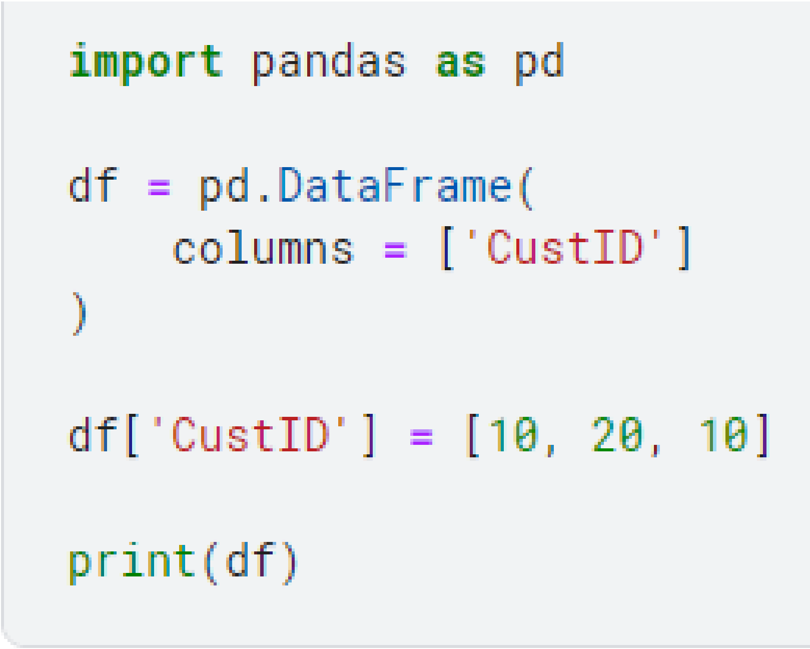
```sql
create table "df" (
"CustID" INTEGER
);

INSERT INTO df values
(10),
(20),
(10);
```



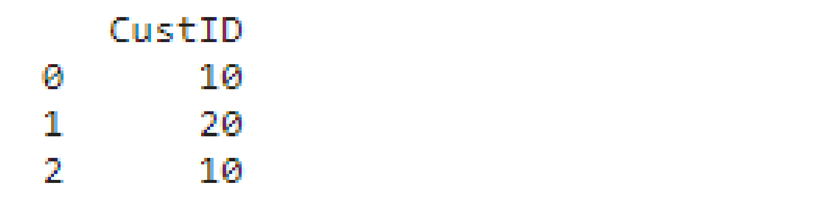
```sql
SELECT COUNT(DISTINCT "CustID")
FROM df;
```



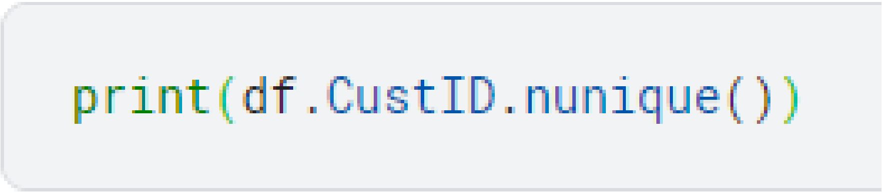
...	COUNT(DISTINCT "CUSTID")
	2



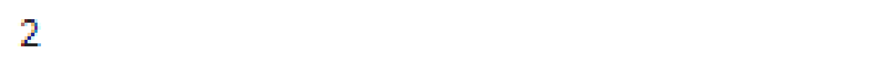
```python
import pandas as pd

df = pd.DataFrame(
    columns = ['CustID']
)

df['CustID'] = [10, 20, 10]

print(df)
```



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



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

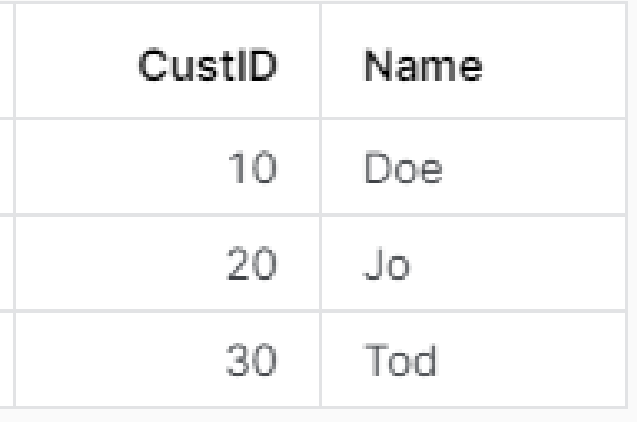


```
2
```

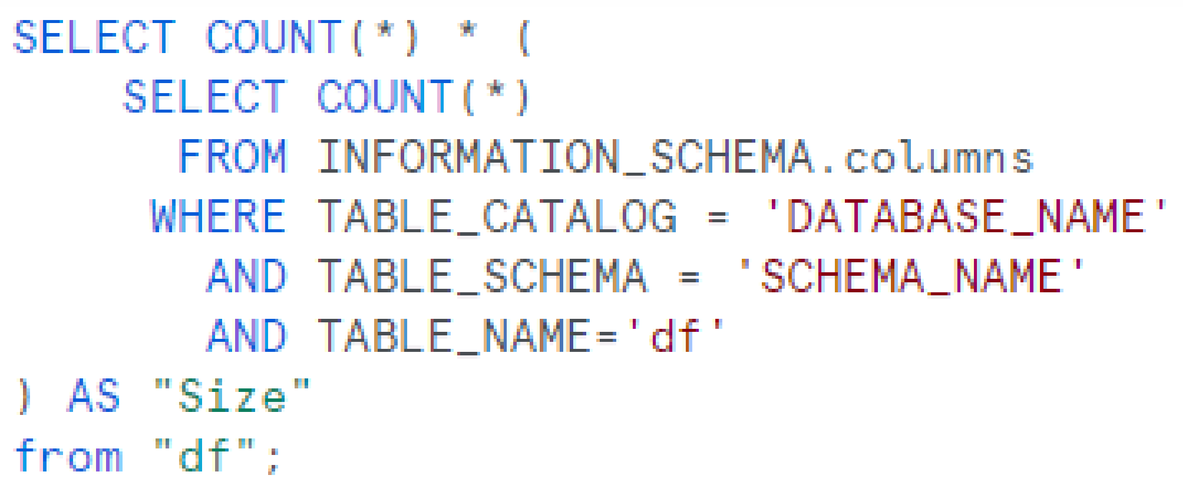
Uzwal Goud Vaddeboina

COUNT OF TOTAL VALUES

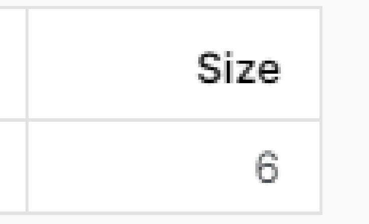
table/dataframe



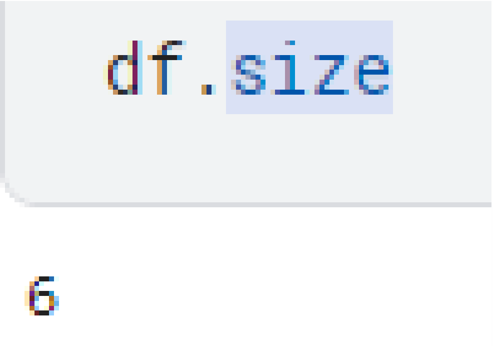
CustID	Name
10	Doe
20	Jo
30	Tod



```
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";
```



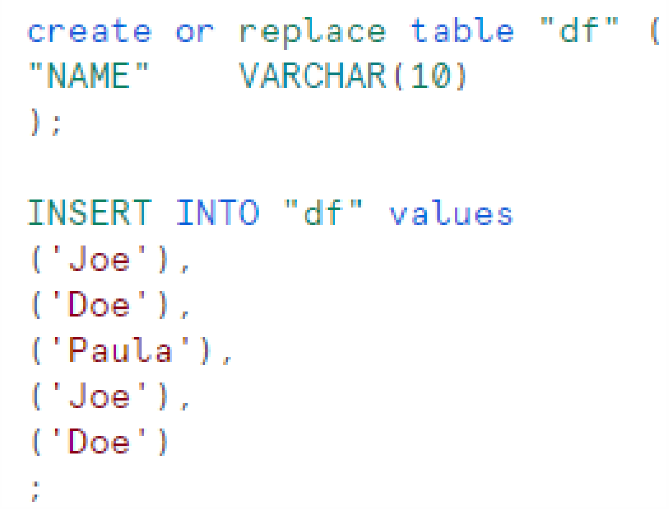
Size
6



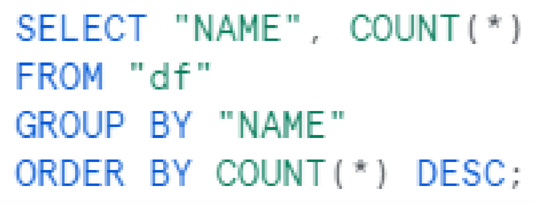
```
df.size
```

```
6
```

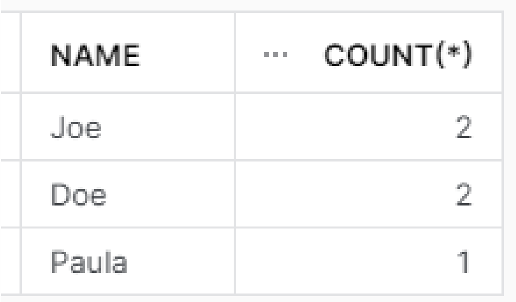
COUNT OF UNIQUE VALUES



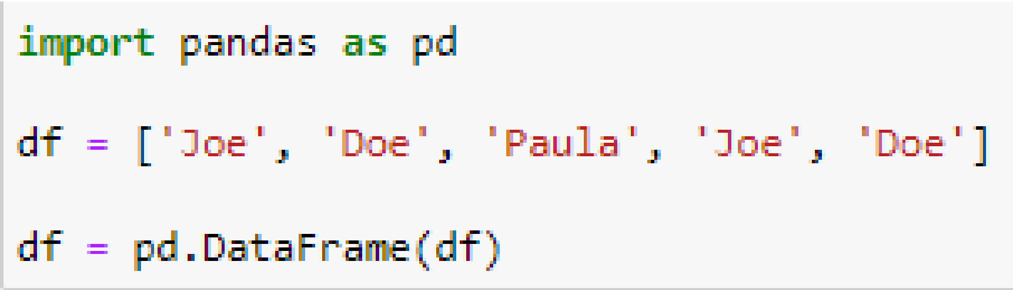
```
create or replace table "df" (
"NAME"    VARCHAR(10)
);

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



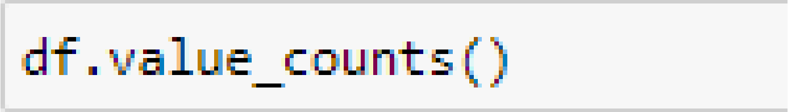
```
SELECT "NAME", COUNT(*)
FROM "df"
GROUP BY "NAME"
ORDER BY COUNT(*) DESC;
```



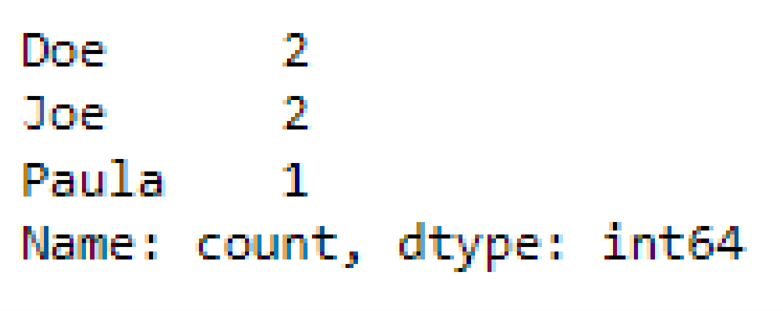
NAME	··· COUNT(*)
Joe	2
Doe	2
Paula	1



```
import pandas as pd

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

df = pd.DataFrame(df)
```



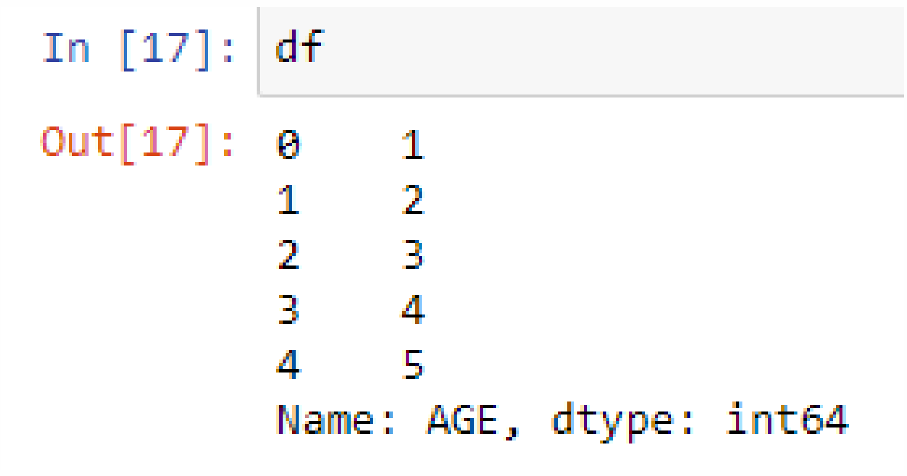
```
df.value_counts()
```



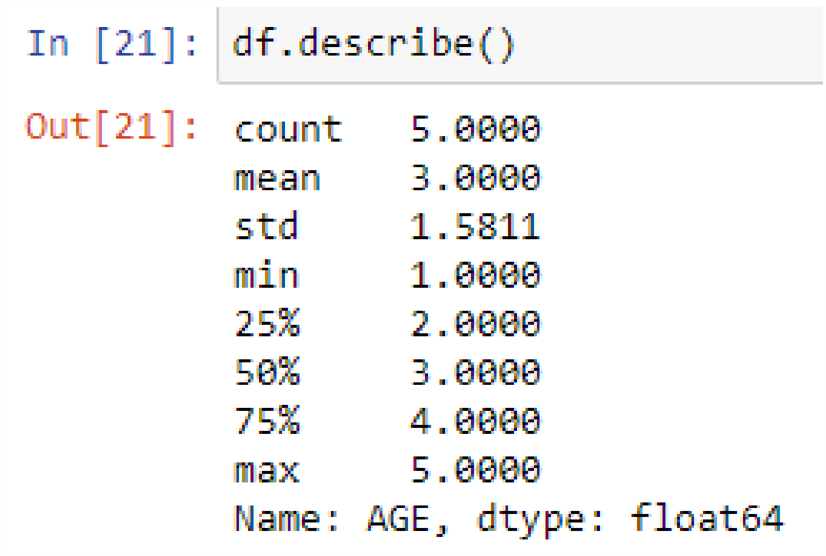
```
Doe       2
Joe       2
Paula     1
Name: count, dtype: int64
```

Uzwal Goud Vaddeboina

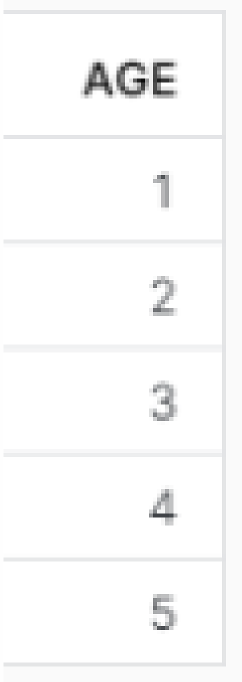
DESCRIPTIVE STATISTICS



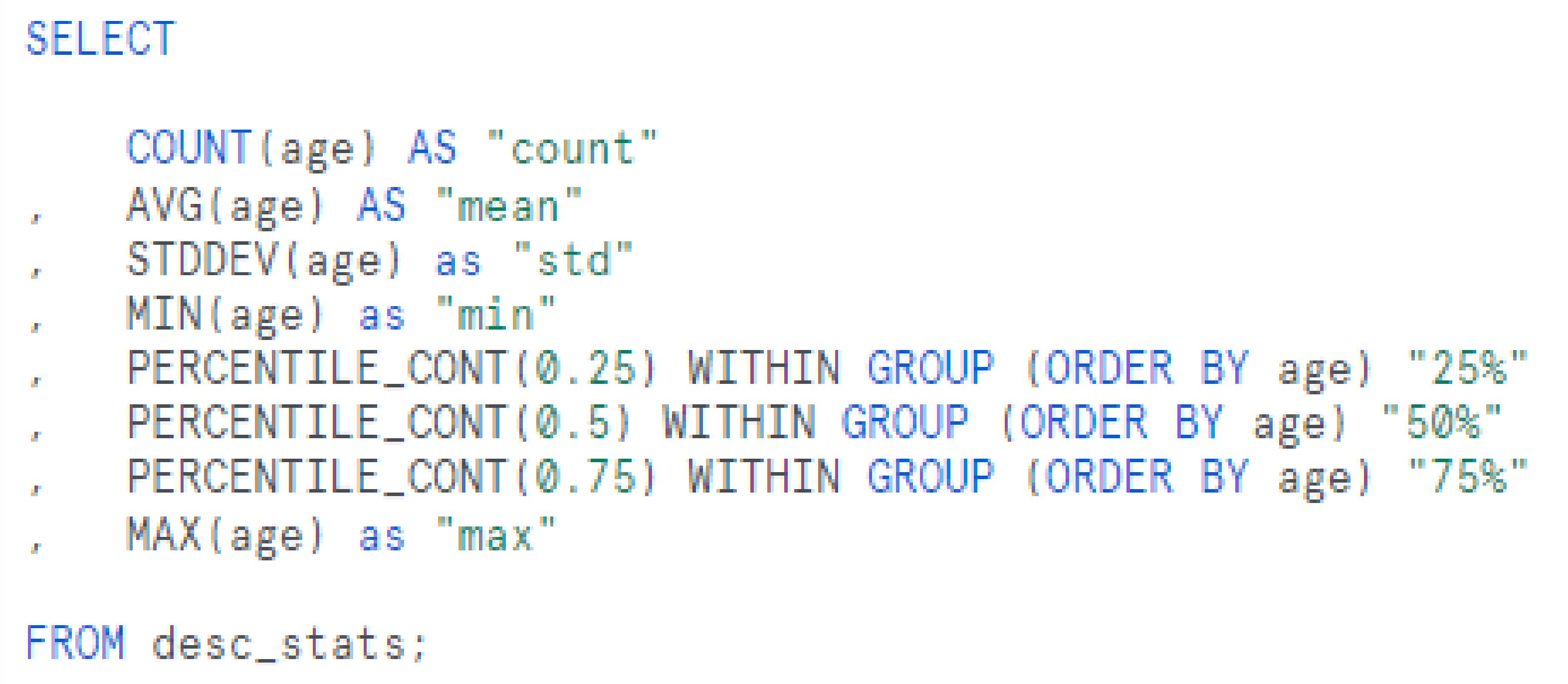
```
In [17]: df

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



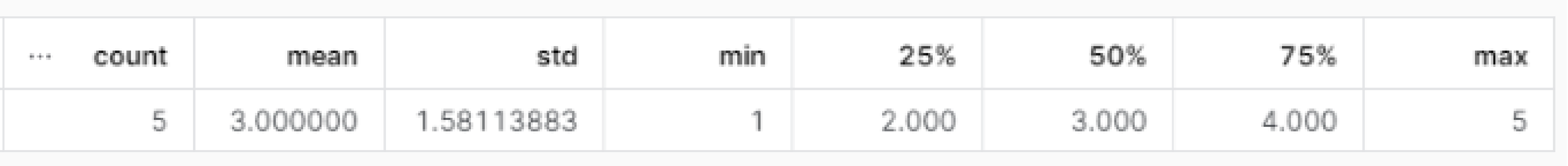
```
In [21]: df.describe()

Out[21]: count    5.0000
         mean     3.0000
         std      1.5811
         min      1.0000
         25%      2.0000
         50%      3.0000
         75%      4.0000
         max      5.0000
         Name: AGE, dtype: float64
```



AGE
1
2
3
4
5



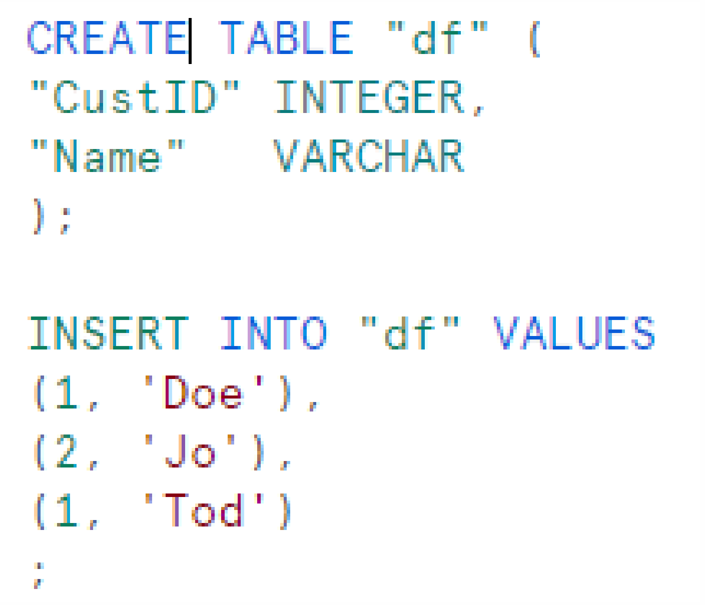
```
SELECT

     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;
```



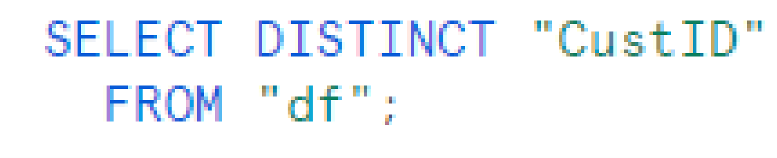
··· count	mean	std	min	25%	50%	75%	max
5	3.000000	1.58113883	1	2.000	3.000	4.000	5

Uzwal Goud Vaddeboina

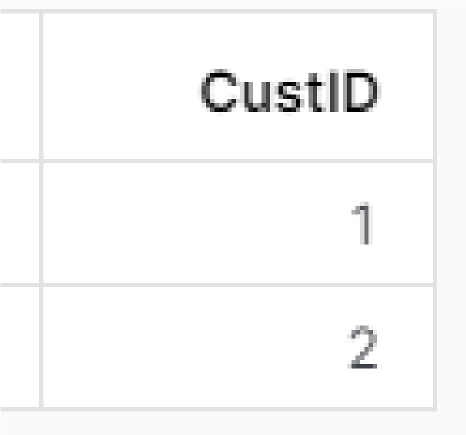
DISTINCT VALUES



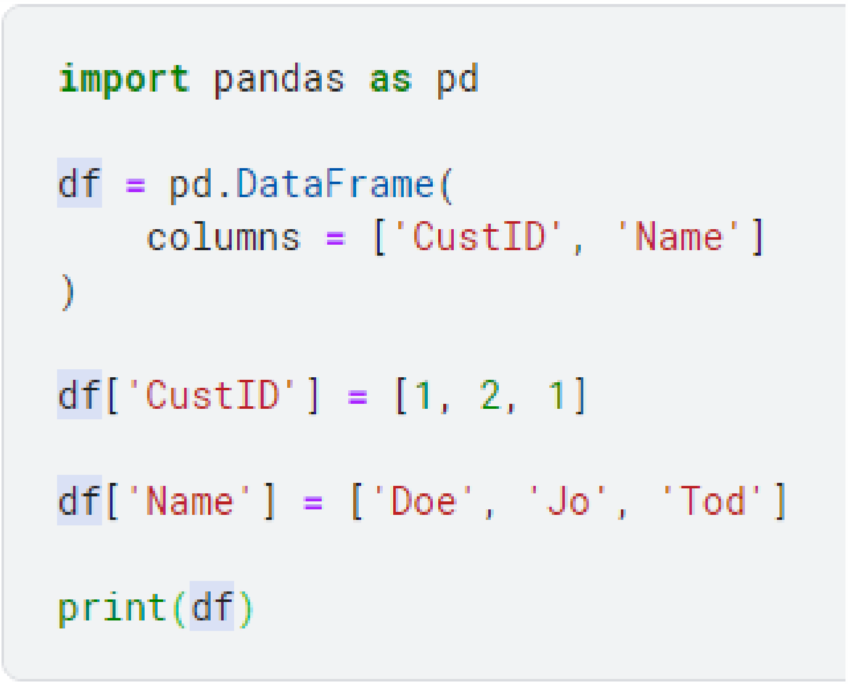
```
CREATE TABLE "df" (
"CustID" INTEGER,
"Name"   VARCHAR
);

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



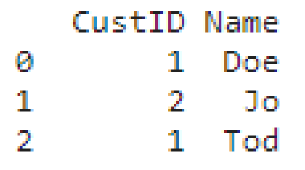
```
SELECT DISTINCT "CustID"
   FROM "df";
```



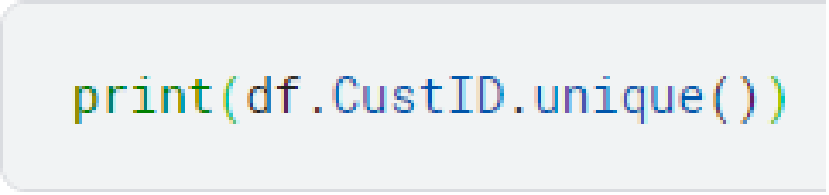
CustID
1
2



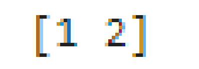
```
import pandas as pd

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
```



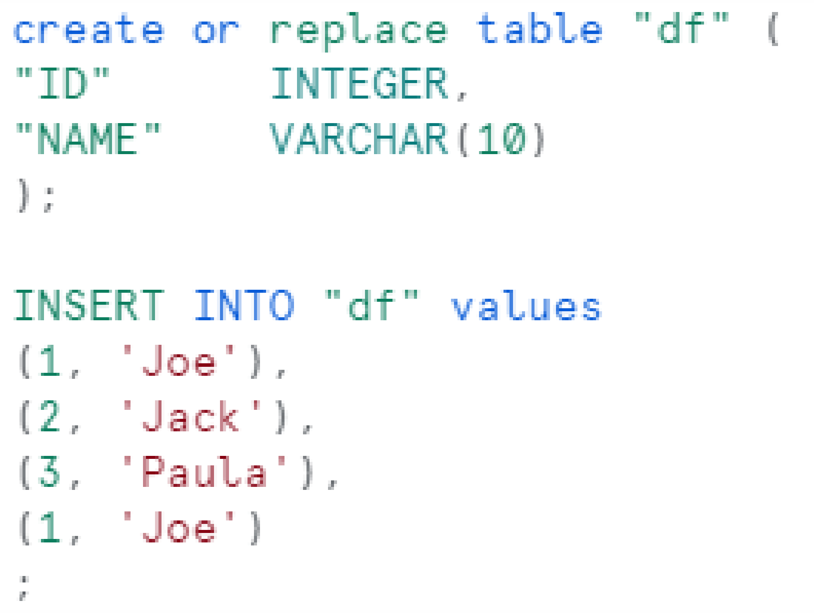
```
print(df.CustID.unique())
```



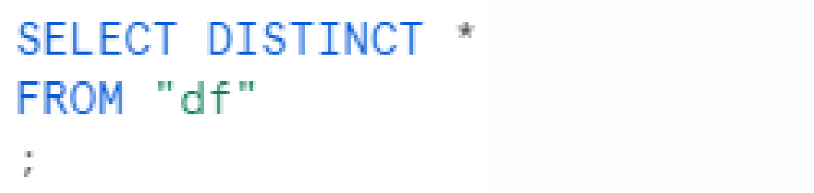
```
[1 2]
```

Uzwal Goud Vaddeboina

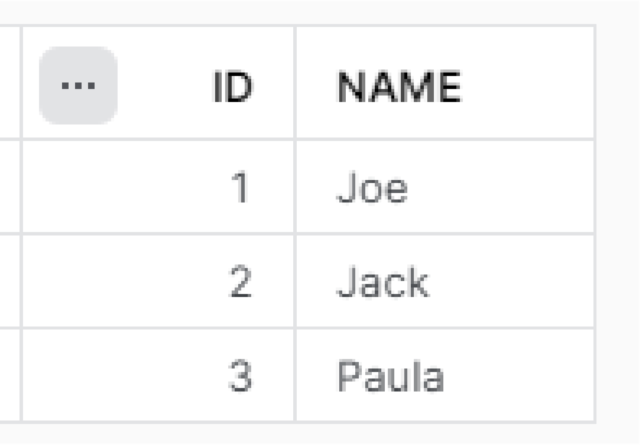
DROP ROW - ALL COLUMNS DUPLICATED



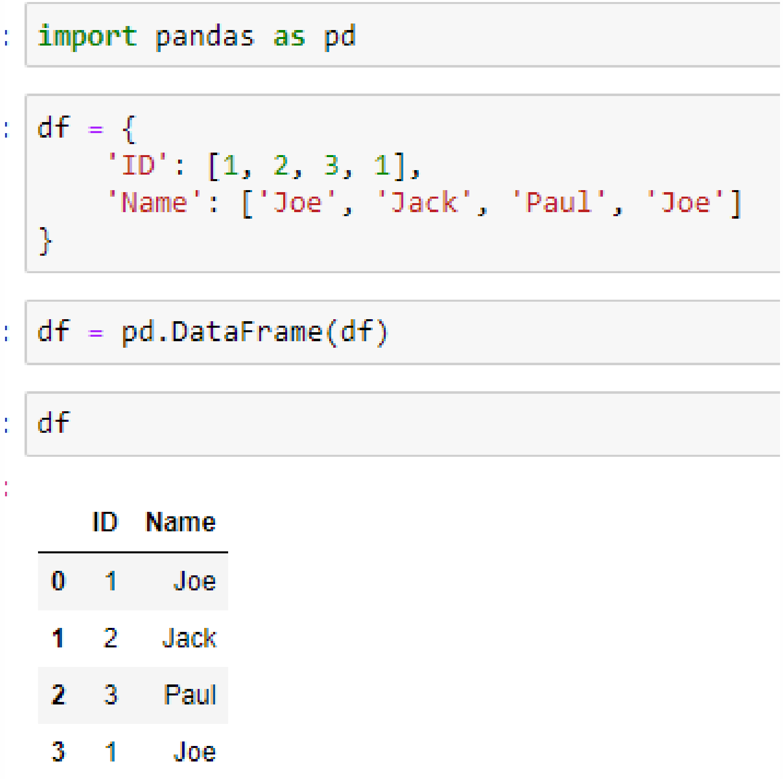
```
create or replace table "df" (
"ID"        INTEGER,
"NAME"      VARCHAR(10)
);

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



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



...	ID	NAME
	1	Joe
	2	Jack
	3	Paula



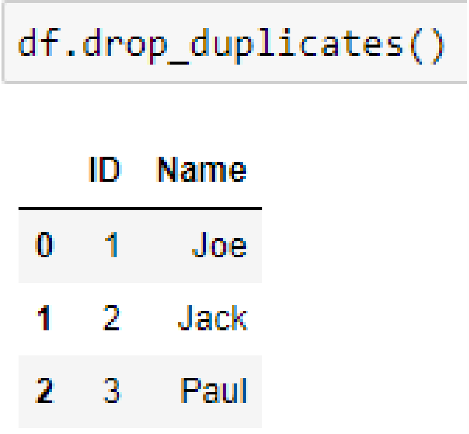
```
import pandas as pd
```

```
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
```

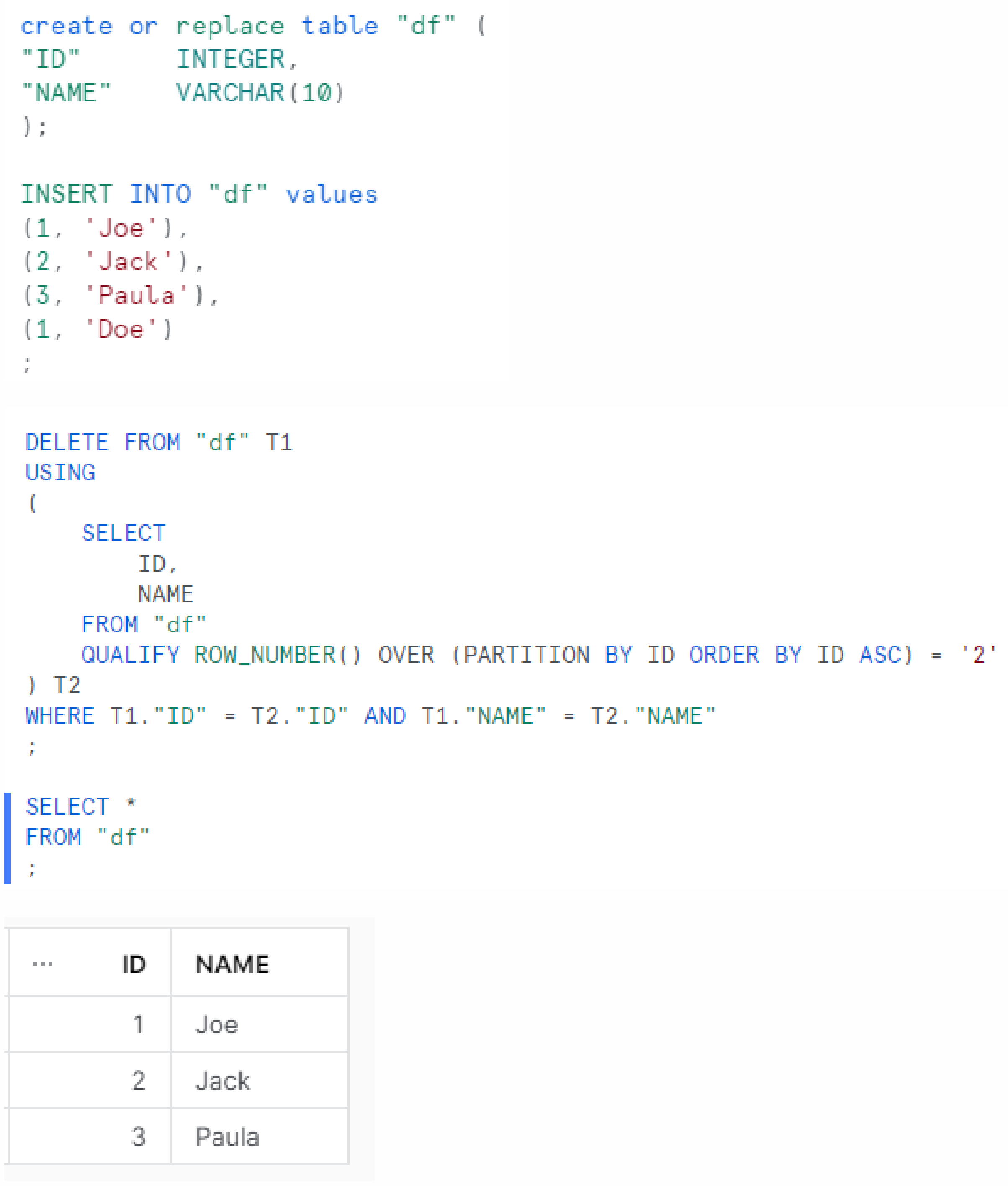


```
df.drop_duplicates()
```

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

Uzwal Goud Vaddeboina

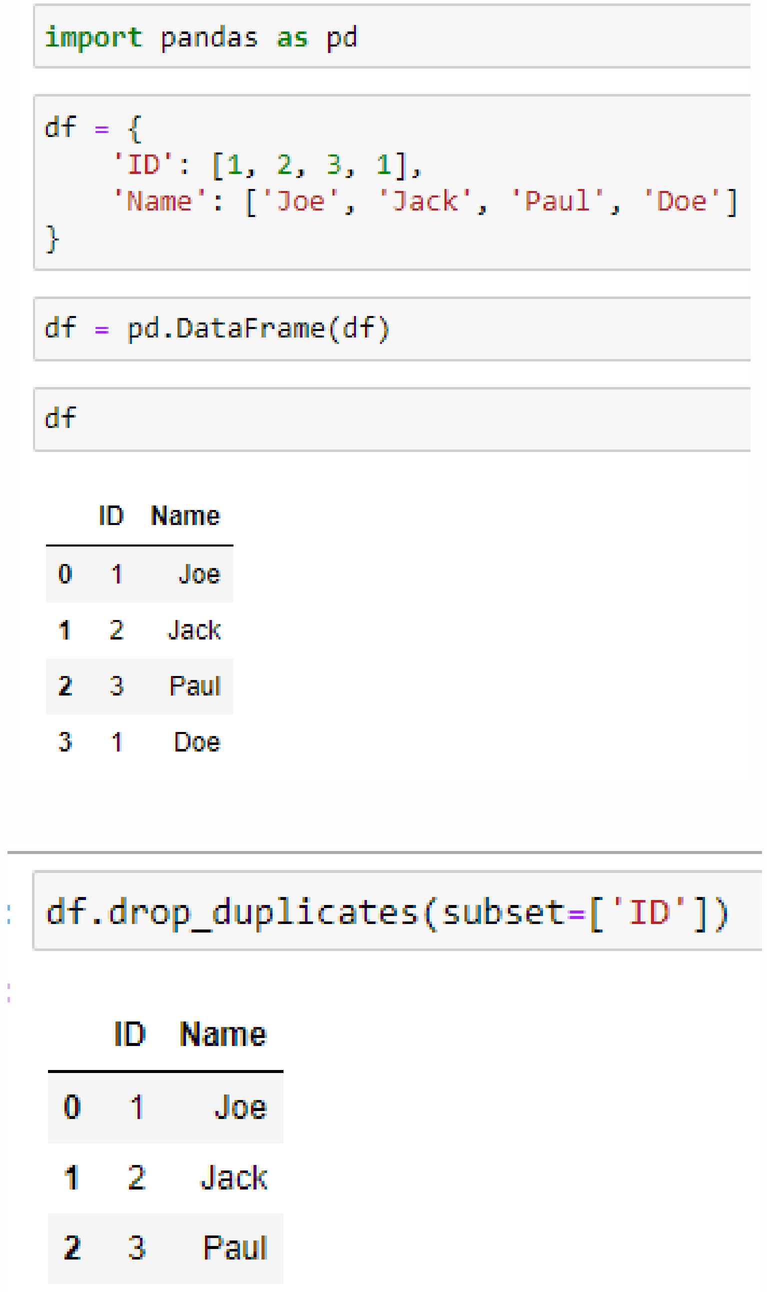
DROP ROW - KEY COLUMN DUPLICATED



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

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

```
DELETE FROM "df" T1
USING
(
    SELECT
        ID,
        NAME
    FROM "df"
    QUALIFY ROW_NUMBER() OVER (PARTITION BY ID ORDER BY ID ASC) = '2'
) T2
WHERE T1."ID" = T2."ID" AND T1."NAME" = T2."NAME"
;

SELECT *
FROM "df"
;
```

...	ID	NAME
	1	Joe
	2	Jack
	3	Paula



```python
import pandas as pd
```

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

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

```python
df
```

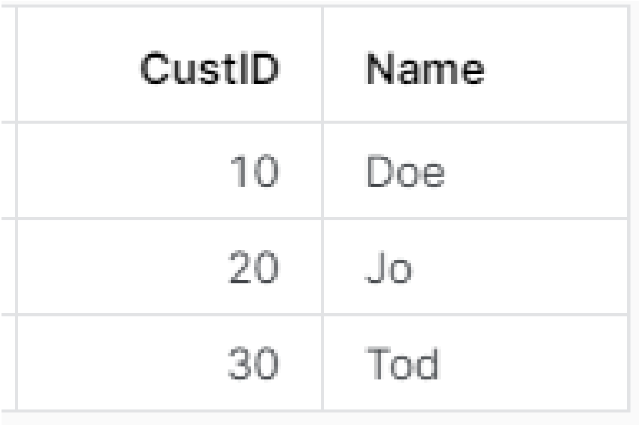
	ID	Name
0	1	Joe
1	2	Jack
2	3	Paul
3	1	Doe

```python
df.drop_duplicates(subset=['ID'])
```

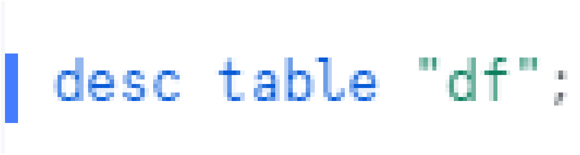
	ID	Name
0	1	Joe
1	2	Jack
2	3	Paul

STRUCTURE OF TABLE

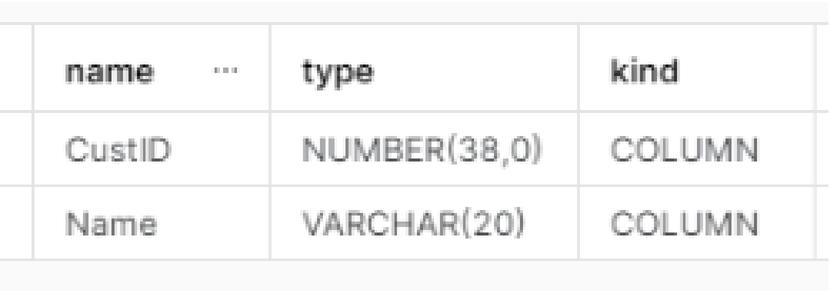
table/dataframe



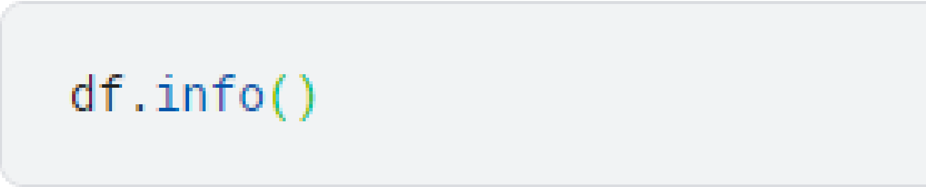
CustID	Name
10	Doe
20	Jo
30	Tod



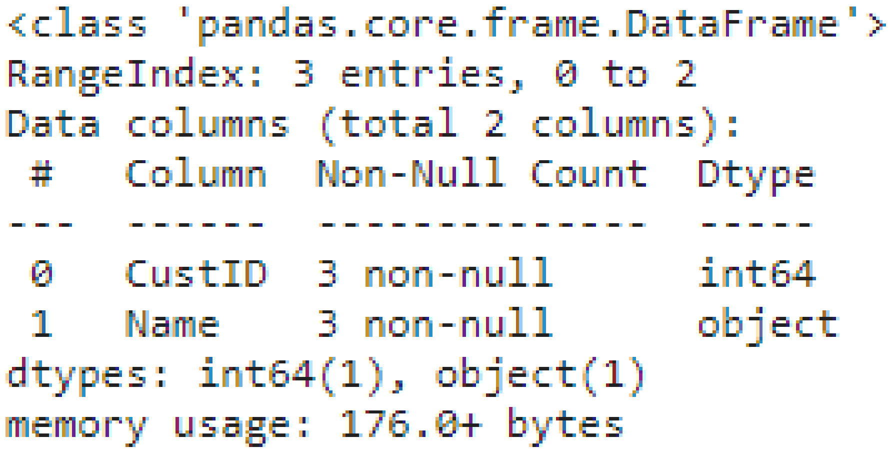
```
desc table "df";
```



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
```

Uzwal Goud Vaddeboina

That's a wrap!

If you liked this content,
follow **Uzwal** on LinkedIn
and click the bell icon
for updates.