

# Merge

module 7

# Merge

- Goal: retrieve information from a table to another

	Program	ProgSkills	Languages	Expert
0	MSIS	4	6.0	1
1	MSIS	3	4.0	1
2	MSIS	3	4.0	1
3	MSIS	3	5.0	1
4	MSIS	3	4.0	1
5	Supply Chain Mgmt & Analytics	1	2.0	0
6	MSIS	3	4.0	1
7	MSIS	2	3.0	1
8	MBA	1	1.0	0
9	MSIS	3	4.0	1

	Program	Units_required
0	MSIS	51
1	MBA	70
2	Master of Finance	48
3	Supply Chain Mgmt & Analytics	49



We want to bring the information on the units required from the table on the right to the table on the left

# Today's data set

- cleaned\_survey.csv

Merge on columns

# df

One row per student =>

```
df[['Program', 'ProgSkills']]
```

	Program	ProgSkills
<b>0</b>	MSIS	4
<b>1</b>	MSIS	3
<b>2</b>	MSIS	3
<b>3</b>	MSIS	3
<b>4</b>	MSIS	3
<b>5</b>	Supply Chain Mgmt & Analytics	1
<b>6</b>	MSIS	3
<b>7</b>	MSIS	2
<b>8</b>	MBA	1
<b>9</b>	MSIS	3

# df\_programs

df\_programs

	Program	Units_required
<b>0</b>	MSIS	51
<b>1</b>	MBA	70
<b>2</b>	Master of Finance	48
<b>3</b>	Supply Chain Mgmt & Analytics	49
<b>4</b>	Master of Hacking	100

One row per Program

A fictitious program



```
df.merge(other_table)
```

Performs the merge on the columns with the same name. In this case,

- `df.Program = df_programs.Program`

## LEFT TABLE

## RIGHT TABLE

```
df.merge(df_programs)
```

[illegible]

# df.merge(other\_table)

*Let's show just few columns*

```
df.merge(df_programs)[['Program', 'ProgSkills', 'Units_required']]
```

	Program	ProgSkills	Units_required
0	MSIS	4	51
1	MSIS	3	51
2	MSIS	3	51
3	MSIS	3	51
4	MSIS	3	51
5	MSIS	3	51



# df.merge(other\_table)

We can specify the keys to merge on for the table on the left (in this case, df) and the table on the right (in this case, df\_programs)

```
df.merge(df_programs, left_on='Program', right_on='Program')
```

	Program	ProgSkills	Units_required
0	MSIS	4	51
1	MSIS	3	51
2	MSIS	3	51
3	MSIS	3	51
4	MSIS	3	51
5	MSIS	3	51

# (default)INNER JOIN: `df.merge(df_programs, left_on='Program', right_on='Program')`

df

	Program	ProgSkills	Languages	Expert
0	MSIS	4	6.0	1
1	MSIS	3	4.0	1
2	MSIS	3	4.0	1
3	MSIS	3	5.0	1
4	MSIS	3	4.0	1
5	Supply Chain Mgmt & Analytics	1	2.0	0
6	MSIS	3	4.0	1
7	MSIS	2	3.0	1
8	MBA	1	1.0	0
9	MSIS	3	4.0	1
16	Faculty!	3	3.0	1
31	Business Man	1	2.0	0

	Program	Units_required
0	MSIS	51
1	MBA	70
2	Master of Finance	48
3	Supply Chain Mgmt & Analytics	49
4	Master of Hacking	100

df\_programs



	Program	ProgSkills	Languages	Expert	Units_required
0	MSIS	4	6.0	1	51
1	MSIS	3	4.0	1	51
2	MSIS	3	4.0	1	51
3	MSIS	3	5.0	1	51
4	MSIS	3	4.0	1	51
5	Supply Chain Mgmt & Analytics	1	2.0	0	49
6	MSIS	3	4.0	1	51
7	MSIS	2	3.0	1	51
8	MBA	1	1.0	0	70
9	MSIS	3	4.0	1	51



## INNER JOIN:

Only the values in both tables are kept:  
“Faculty!” and “Business Man” from df and “Master of Hacking” from df\_programs are dropped

# LEFT JOIN:


```
df.merge(df_programs, left_on='Program', right_on='Program', how='left')
```

df

	Program	ProgSkills	Languages	Expert
0	MSIS	4	6.0	1
1	MSIS	3	4.0	1
2	MSIS	3	4.0	1
3	MSIS	3	5.0	1
4	MSIS	3	4.0	1
5	Supply Chain Mgmt & Analytics	1	2.0	0
6	MSIS	3	4.0	1
7	MSIS	2	3.0	1
8	MBA	1	1.0	0
9	MSIS	3	4.0	1
16	Faculty!	3	3.0	1
31	Business Man	1	2.0	0

	Program	Units_required
0	MSIS	51
1	MBA	70
2	Master of Finance	48
3	Supply Chain Mgmt & Analytics	49
4	Master of Hacking	100

df\_programs



	Program	ProgSkills	Languages	Expert	Units_required
0	MSIS	4	6.0	1	51
1	MSIS	3	4.0	1	51
2	MSIS	3	4.0	1	51
3	MSIS	3	5.0	1	51
4	MSIS	3	4.0	1	51
5	Supply Chain Mgmt & Analytics	1	2.0	0	49
6	MSIS	3	4.0	1	51
7	MSIS	2	3.0	1	51
8	MBA	1	1.0	0	70
9	MSIS	3	4.0	1	51
16	Faculty!	3	3.0	1	NaN
31	Business Man	1	2.0	0	NaN

## LEFT JOIN:

All values from the left table are kept:  
“Faculty!” and “Business Man” are kept, “Master of Hacking” is not

# OUTER JOIN:

```
df.merge(df_programs, left_on='Program', right_on='Program', how='outer')
```

df

	Program	ProgSkills	Languages	Expert
0	MSIS	4	6.0	1
1	MSIS	3	4.0	1
2	MSIS	3	4.0	1
3	MSIS	3	5.0	1
4	MSIS	3	4.0	1
5	Supply Chain Mgmt & Analytics	1	2.0	0
6	MSIS	3	4.0	1
7	MSIS	2	3.0	1
8	MBA	1	1.0	0
9	MSIS	3	4.0	1
16	Faculty!	3	3.0	1
31	Business Man	1	2.0	0

	Program	Units_required
0	MSIS	51
1	MBA	70
2	Master of Finance	48
3	Supply Chain Mgmt & Analytics	49
4	Master of Hacking	100

df\_programs



	Program	ProgSkills	Languages	Expert	Units_required
0	MSIS	4	6.0	1	51
1	MSIS	3	4.0	1	51
2	MSIS	3	4.0	1	51
3	MSIS	3	5.0	1	51
4	MSIS	3	4.0	1	51
5	Supply Chain Mgmt & Analytics	1	2.0	0	49
6	MSIS	3	4.0	1	51
7	MSIS	2	3.0	1	51
8	MBA	1	1.0	0	70
9	MSIS	3	4.0	1	51
16	Faculty!	3	3.0	1	NaN
31	Business Man	1	2.0	0	NaN
61	Master of Hacking	NaN	NaN	NaN	100.0

## OUTER JOIN:

All values from the both tables are kept:

“Faculty!” and “Business Man” are kept, as well as

“Master of Hacking”

Merge on indices

# df

One row per student =>

```
df[['Program', 'ProgSkills']]
```

	Program	ProgSkills
<b>0</b>	MSIS	4
<b>1</b>	MSIS	3
<b>2</b>	MSIS	3
<b>3</b>	MSIS	3
<b>4</b>	MSIS	3
<b>5</b>	Supply Chain Mgmt & Analytics	1
<b>6</b>	MSIS	3
<b>7</b>	MSIS	2
<b>8</b>	MBA	1
<b>9</b>	MSIS	3

# df\_programs\_i

```
df_programs_i = df_programs.set_index('Program')
```

df\_programs\_i

One row per Program

**Program is the index**

	Units_required
Program	
MSIS	51
MBA	70
Master of Finance	48
Supply Chain Mgmt & Analytics	49
Master of Hacking	100

The key to use in the right table is the index

```
df.merge(df_programs_i, left_on = 'Program', right_index=True)
```

	Program	ProgSkills	Units_required
0	MSIS	4	51
1	MSIS	3	51
2	MSIS	3	51
3	MSIS	3	51
4	MSIS	3	51
6	MSIS	3	51
7	MSIS	2	51



# Problems

1. For each programming skills level, find the average number of units to be completed by students with that programming skill level
2. For each existing program (i.e., for each Program in df\_programs), find the units required to complete it and the number of students belonging to that program that responded to the survey.
3. For each student in df\_students, the number of weekly hours they are working, assuming that:
  1. each required unit of coursework is 0.25 hours a week of work
  2. Job=0 is 0 hours a week of work
  3. Job=0.5 is 20 hours a week of work
  4. Job=1 is 40 hours a week of work