Pandas + SQL

Tidy Data

This Lecture

Some more thoughts on Pandas and SQL

Before we start...

Before we start...

1. Say hello to the mods

Before we start...

1. Say hello to the mods
2. Some updates

Today's Moderator

Today's Moderator

1. Laura!

Today's Moderator

1. Laura!
2. Currently a first-year grad student at UMD

Today's Moderator

1. Laura!
2. Currently a first-year grad student at UMD
3. ...also did her undergrad at UMD

Some updates

Some updates

1. New docker image!

Some updates

1. New docker image!
2. Prettify is making things ugly

Making DataFrames from the web

On Tuesday I was asked about this and said that I wouldn't be showing an example, but I can show an example today!

Making DataFrames from the web

On Tuesday I was asked about this and said that I wouldn't be showing an example, but I can show an example today!
To the Notebook

Melting?!?!

Some students were confused about Melting (from the reading, you should be doing the reading!)

Melting?!?!

Some students were confused about Melting (from the reading, you should be doing the reading!) I made a Notebook to explain!

SQL

One of the most important DSLs.

SQL

One of the most important DSLs.

1. SQL: Working with *structured* data

SQL

One of the most important DSLs.

1. SQL: Working with *structured* data
2. More specifically: relational data

SQL

One of the most important DSLs.

1. SQL: Working with *structured* data
2. More specifically: relational data
3. As a rough guide: Think tuples!

SQL vs Pandas

Many pixels have been used in this debate. My take:

SQL vs Pandas

Many pixels have been used in this debate. My take:

1. If you have a choice, choose the one you're most comfortable with

SQL vs Pandas

Many pixels have been used in this debate. My take:

1. If you have a choice, choose the one you're most comfortable with
2. If you don't have a choice, you use the one you have to

SQL vs Pandas

Many pixels have been used in this debate. My take:

1. If you have a choice, choose the one you're most comfortable with
2. If you don't have a choice, you use the one you have to
3. There are many situations where the choice is beyond your control (i.e. not just because a manager makes you)

SQL

Tables are important!

SQL

Tables are important!

ID	Age	Weight (kg)	Height (cm)
1	12.2	42.3	146.1
2	11.0	40.8	143.8
3	15.6	65.3	165.3
4	35.1	84.2	185.8

SQL

Tables are important!

ID	Age	Weight (kg)	Height (cm)
1	12.2	42.3	146.1
2	11.0	40.8	143.8
3	15.6	65.3	165.3
4	35.1	84.2	185.8

We've got Labels, Variables, and Observations

SQL: Primary Keys

Primary Keys: Unique identifier for one observation in one table.

SQL: Primary Keys

Primary Keys: Unique identifier for one observation in one table.

ID	Age	Weight (kg)	Height (cm)	Nationality
1	12.2	42.3	146.1	1
2	11.0	40.8	143.8	2
3	15.6	65.3	165.3	5
4	35.1	84.2	185.8	3

SQL: Primary Keys

Primary Keys: Unique identifier for one observation in one table.

ID	Age	Weight (kg)	Height (cm)	Nationality
1	12.2	42.3	146.1	1
2	11.0	40.8	143.8	2
3	15.6	65.3	165.3	5
4	35.1	84.2	185.8	3

When the primary key *from another table* is used in a table, like above, we call them foreign keys

High-level view: Cons

SQL is very powerful, but has some downsides

High-level view: Cons

SQL is very powerful, but has some downsides

1. You have to think about the 'schema' beforehand

High-level view: Cons

SQL is very powerful, but has some downsides

1. You have to think about the 'schema' beforehand
2. Some people don't like SQL (the language)

High-level view: Cons

SQL is very powerful, but has some downsides

1. You have to think about the ‘schema’ beforehand
2. Some people don’t like SQL (the language)
3. YATTL :(

High-level view: Pros

SQL is very powerful, more power is more gooder

High-level view: Pros

SQL is very powerful, more power is more gooder

1. Implementations are *very* good

High-level view: Pros

SQL is very powerful, more power is more gooder

1. Implementations are *very* good
2. Schemas are important, thinking about them ahead of time is good!

High-level view: Pros

SQL is very powerful, more power is more gooder

1. Implementations are *very* good
2. Schemas are important, thinking about them ahead of time is good!
3. You can share data with folks across different tech stacks!

High-level view: Pros

SQL is very powerful, more power is more gooder

1. Implementations are *very* good
2. Schemas are important, thinking about them ahead of time is good!
3. You can share data with folks across different tech stacks!
4. The same system can process small, medium, large, huge amounts of data.

High-level view: Implementations

SQL is very powerful, but it's not one thing

High-level view: Implementations

SQL is very powerful, but it's not one thing

1. SQLite: The most popular database ever. By far., great for small/medium datasets

High-level view: Implementations

SQL is very powerful, but it's not one thing

1. SQLite: The most popular database ever. By far., great for small/medium datasets
2. MySQL/MariaDB: Lots of documentation, very common for the web.

High-level view: Implementations

SQL is very powerful, but it's not one thing

1. SQLite: The most popular database ever. By far., great for small/medium datasets
2. MySQL/MariaDB: Lots of documentation, very common for the web.
3. PostgreSQL: Fantastic for large amounts of data, 'industrial strength'

Thanks for your time!