SQL & Python for Analytics

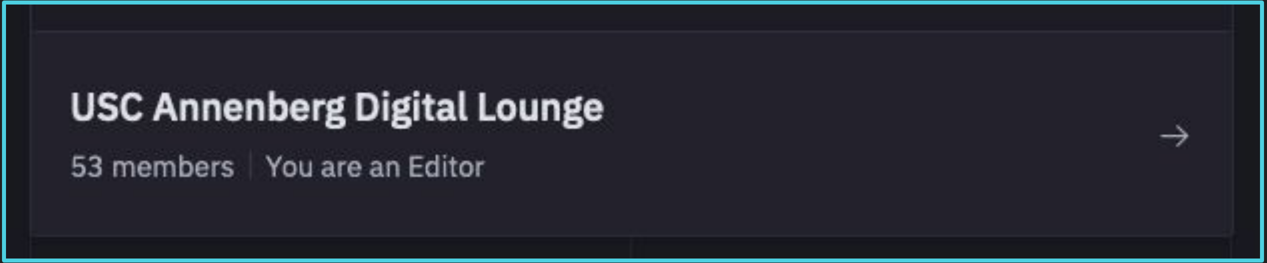
USC Annenberg Digital Lounge
Week 2: Data Modeling with SQL

Today’s session

This is going to be interactive, so please be ready to try out the exercises as we go!

We are going to be using a tool called Hex, which is a tool for using SQL and Python for doing data analysis and visualization. Shout-out to Hex for giving us their Professional Plan for free!

Go to app.hex.tech and sign up with your USC email address or Microsoft login. When asked to select a workspace, choose Annenberg Digital Lounge:

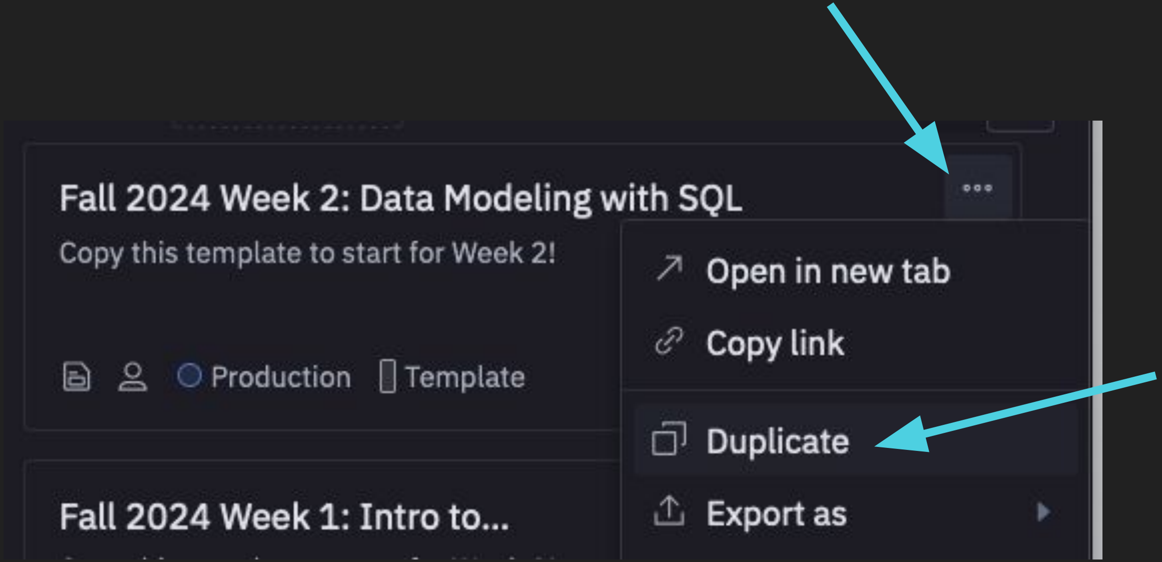


USC Annenberg Digital Lounge

53 members | You are an Editor

Your workbook

In Hex, look for "**Fall 2024 Week 2**", then click the "**...**" next to the name, then click **Duplicate**.


Fall 2024 Week 2: Data Modeling with SQL
Copy this template to start for Week 2!
Production
Template
Open in new tab
Copy link
Duplicate
Export as
Fall 2024 Week 1: Intro to...

Workshop overview

- Week 1: Intro to SQL
- Week 2 (Today!): Data Modeling with SQL
- Week 3 (Nov. 18): Intro to Python
- Week 4 (Nov. 25): Analyzing Data with Python
- Week 5 (Dec. 2): Analytics & Visualization with SQL and Python

Today’s agenda

- Review SQL 101
- Data Modeling
- Joining Data Frames
- Casting

SQL Review

SELECT (columns, or a function of columns, or * for all)

FROM (table)

JOIN (other table)

WHERE (conditions)

GROUP BY (columns)

ORDER BY (which column)

LIMIT (if you only want a certain number of results)

Sometimes those words are written in all caps by convention, but it is not required

Our First Queries

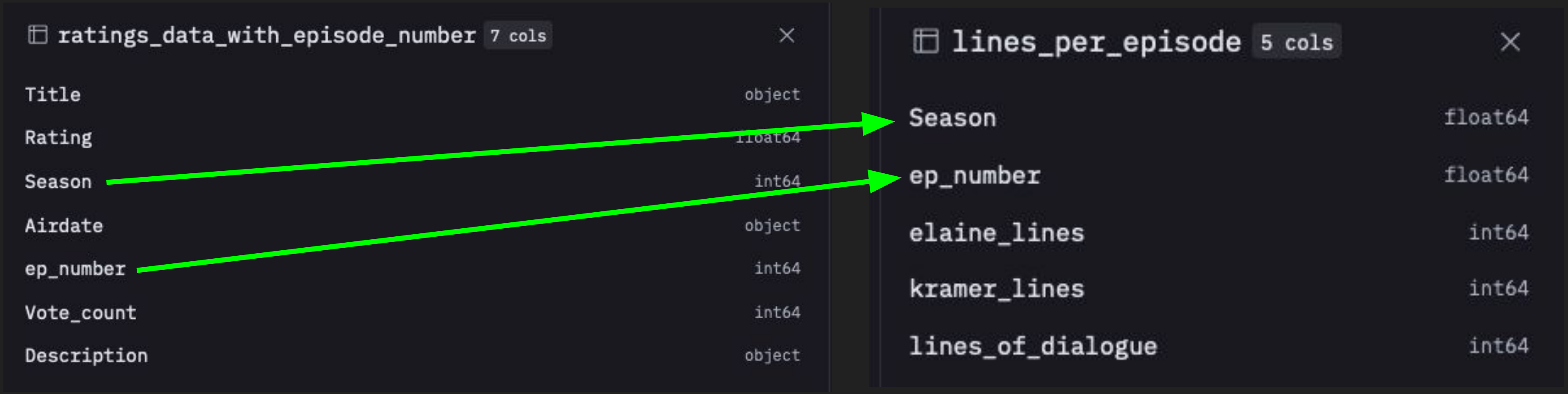
```
select * from transcripts_data limit 20


select distinct character,
count(*) as number_of_lines
from transcripts_data
group by character
order by number_of_lines desc
```

Our First Queries - Formulas

```sql
select season,
"EpisodeNo" as ep_number,
count("Dialogue") as lines_of_dialogue,
count(case when character = 'ELAINE' then "Dialogue" end) as elaine_lines,
count(case when character = 'KRAMER' then "Dialogue" end) as kramer_lines
from transcripts_data
group by season, "EpisodeNo"
```

Our First Queries – Joining Dataframes

```
select * from ratings_data_with_episode_number
join (select * from lines_per_episode) using (season, ep_number)
```



ratings_data_with_episode_number 7 cols

Title	object
Rating	float64
Season	int64
Airdate	object
ep_number	int64
Vote_count	int64
Description	object

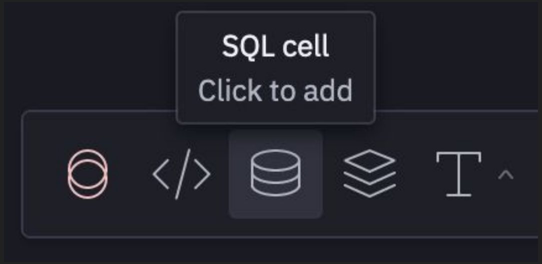
lines_per_episode 5 cols

Season	float64
ep_number	float64
elaine_lines	int64
kramer_lines	int64
lines_of_dialogue	int64

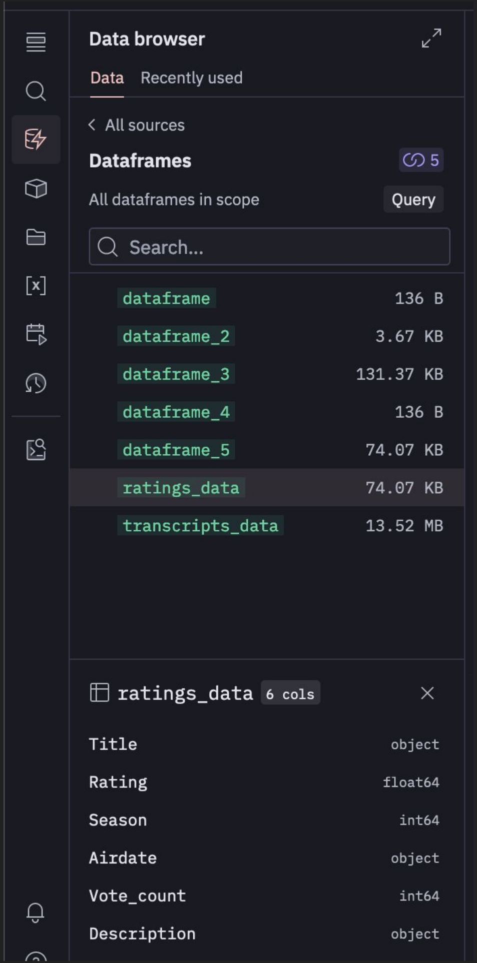
Navigating Hex

- How do I know what columns are available to select from?
- Go to Data Browser → choose a dataframe
- See all the columns listed below

Add SQL cells



Command-Enter to run a cell (and any dependent cells)



dataframe	136 B
dataframe_2	3.67 KB
dataframe_3	131.37 KB
dataframe_4	136 B
dataframe_5	74.07 KB
ratings_data	74.07 KB
transcripts_data	13.52 MB

ratings_data 6 cols

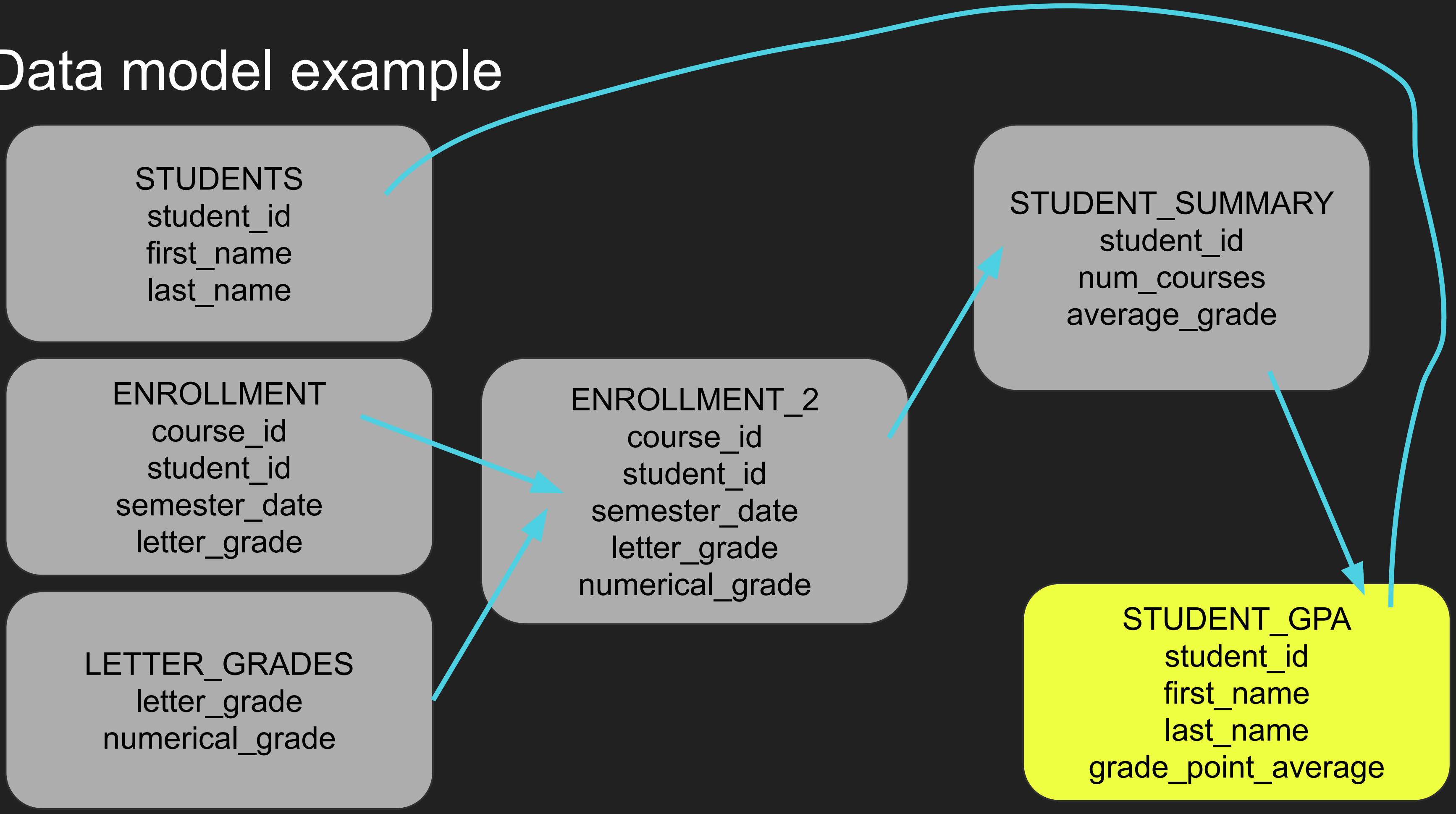
Title	object
Rating	float64
Season	int64
Airdate	object
Vote_count	int64
Description	object

Time for exercise 1!

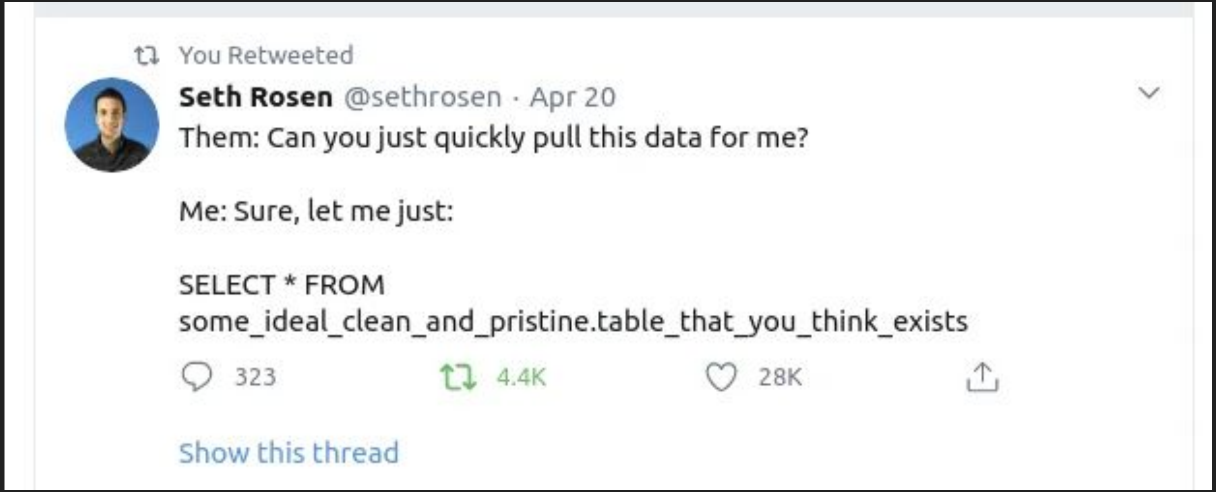
What is data modeling?

- A set of SQL queries that “clean up” your source data, to make it easy to ask and answer business questions of your data
- Your initial datasets are “source data” – these come from systems
- Your completed data models take that source data, do any joins or formulas that your end users might commonly need to do for their queries

Data model example



Why data modeling?


You Retweeted

Seth Rosen @sethrosen · Apr 20

Them: Can you just quickly pull this data for me?

Me: Sure, let me just:

SELECT * FROM
some_ideal_clean_and_pristine.table_that_you_think_exists

323 4.4K 28K

Show this thread

Types of Joins

Regular Join

- All rows from both dataframes that have matching values
- If any row in either dataframe doesn’t have a match in the other, it will be dropped

Left Join

- ALL the rows from the left dataframe, plus anything in the right dataframe that has matching values

Steps for joining datasets

Figure out…	**Build your query**

What tables have the source columns you need?	SELECT those_columns
	FROM that_table

Does each table have one row per entity? If not, summarize first so that they do in a new dataframe	SELECT sport, count(athlete_id) GROUP BY 1
Do you need only rows that appear in both tables? Or all rows from one table?	JOIN table table_name LEFT JOIN table table_name
For all the tables you need, what column that has a unique identifier that you can use to match records across tables?	USING (same_column_title) ON column1=column2

What columns do you want to end up with? Do you need any formulas?	SELECT those_columns,
	a_formula(a_column) AS result
	FROM that_table

WHERE Conditions

Can also be used in the WHEN part of CASE statements!

Numbers

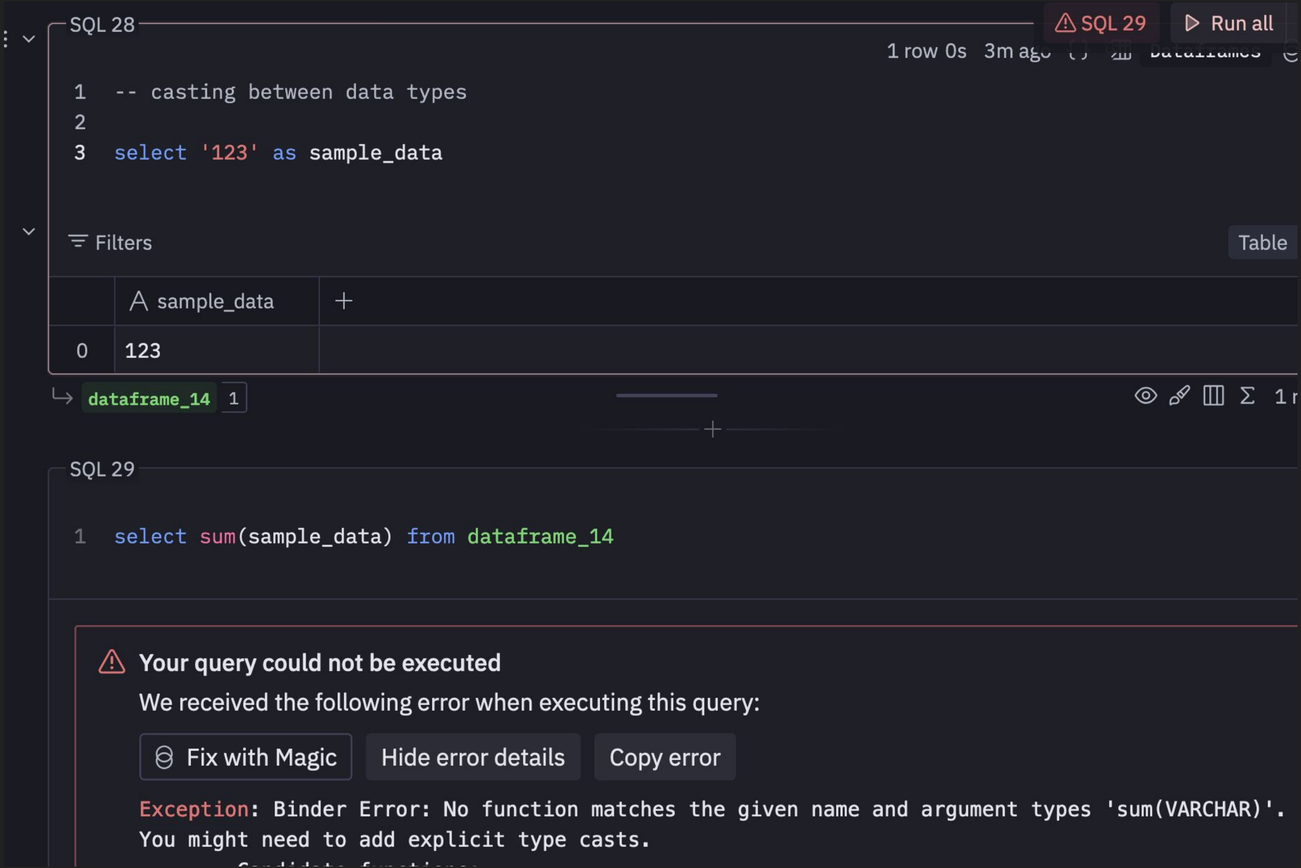
- = equal
- != is not equal
- < less than
- > greater than
- <= less than or equal to
- >= greater than or equal to

Strings/Text

- = equal
- != is not equal
- IN (‘othertext1’, ‘othertext2’)
- LIKE ‘this exact TEXt’
- ILIKE ‘%this phrase%’

Time for exercise 2!

Casting Data Types



SQL 28

```sql
-- casting between data types

select '123' as sample_data
```

	sample_data
0	123

dataframe_14 1

SQL 29

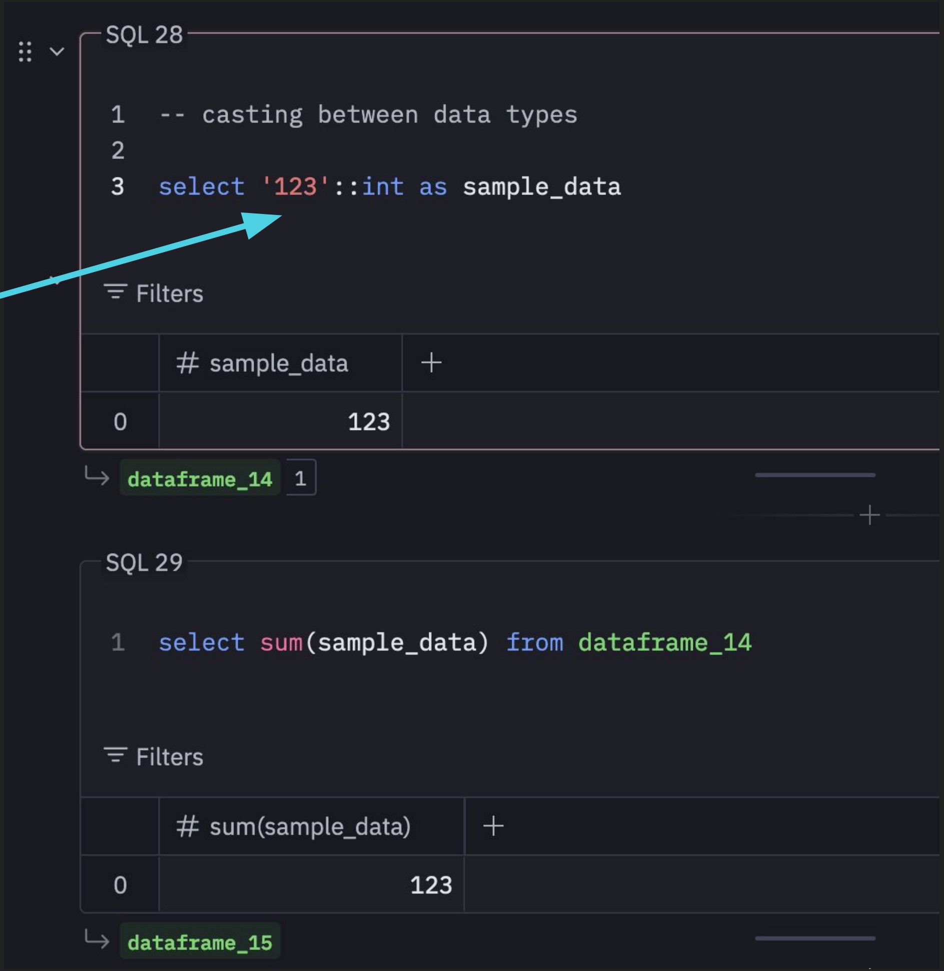
```sql
select sum(sample_data) from dataframe_14
```

Your query could not be executed

We received the following error when executing this query:

```
Exception: Binder Error: No function matches the given name and argument types 'sum(VARCHAR)'.
You might need to add explicit type casts.
```

Casting Data Types



SQL 28

```sql
-- casting between data types

select '123'::int as sample_data
```

Filters

	# sample_data
0	123

dataframe_14 1

SQL 29

```sql
select sum(sample_data) from dataframe_14
```

Filters

	# sum(sample_data)
0	123

dataframe_15

Keep learning!

Upcoming sessions:

- Week 3 (Nov. 18): Intro to Python
- Week 4 (Nov. 25): Analyzing Data with Python
- Week 5 (Dec. 2): Analytics & Visualization with SQL and Python

Good tutorials at:

- Data modeling: docs.getdbt.com/best-practices/how-we-style/1-how-we-style-our-dbt-models
- learn.hex.tech
- Learn SQL via a Murder Mystery: https://mystery.knightlab.com/
- sqlbolt.com for a more standard tutorial