tinyurl.com/jc25-data-eng

JupyterCon 2025

Teaching Data Engineering at Scale With Jupyter Notebooks at UC Berkeley

November 5, 2025

Rebecca Dang, Jonathan M. Ferrari, Christy Quang, Michael Ball, and Lisa Yan

CC BY NC SA

How do you use Jupyter to teach data engineering to a course of over **400 students** each semester and with **fewer than 12 course staff** members?

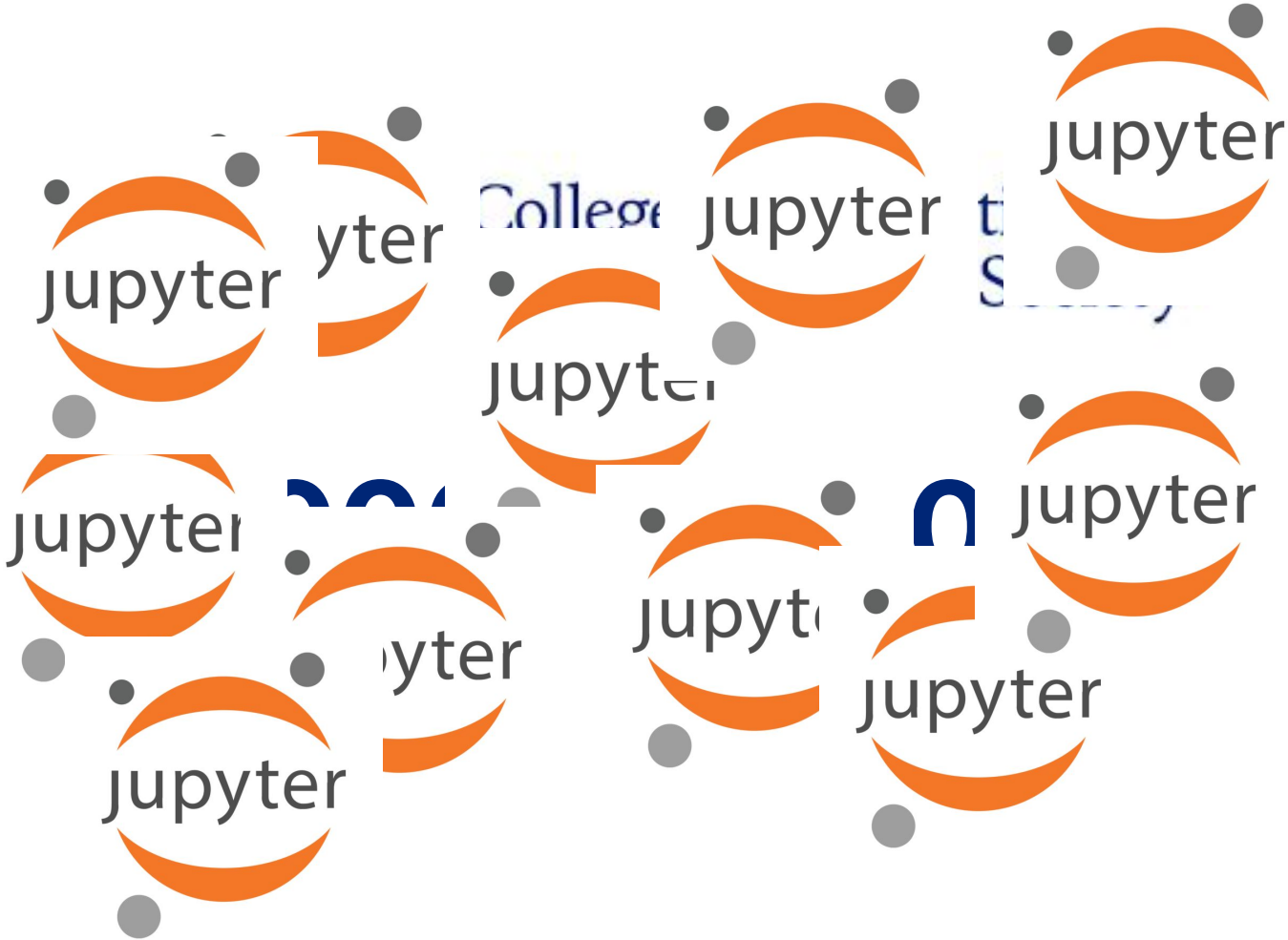
CC BY NC SA

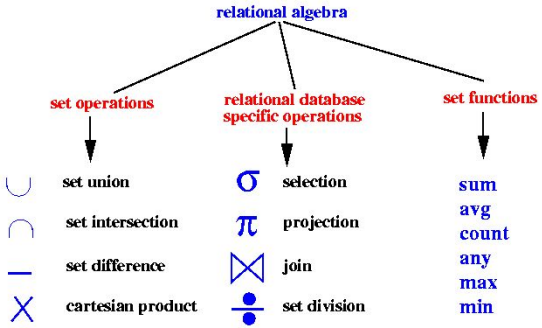
UC Berkeley College of Computing, Data Science, and Society

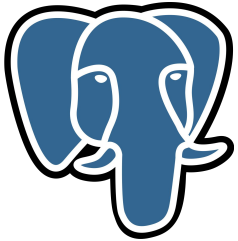
2000
students

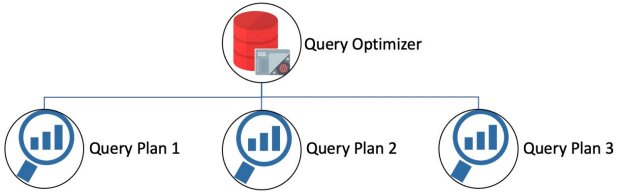
10
classes

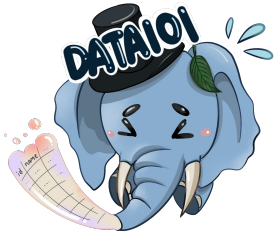
CC BY NC SA

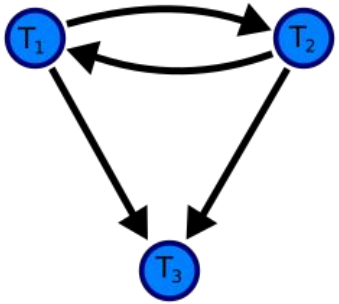

jupyter
jupyter
jupyter
jupyter
jupyter
jupyter
jupyter
jupyter
jupyter


relational algebra
set operations
relational database specific operations
set functions
∪ set union
∩ set intersection
− set difference
X cartesian product
σ selection
π projection
⋈ join
÷ set division
sum
avg
count
any
max
min

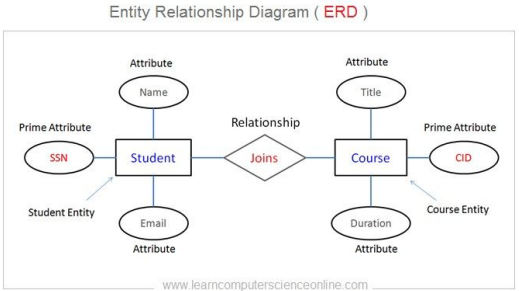



Query Optimizer
Query Plan 1
Query Plan 2
Query Plan 3


DATA101

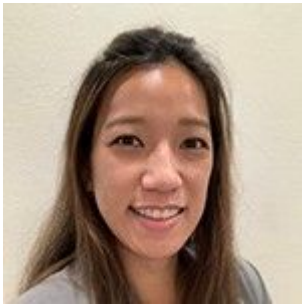

T1
T2
T3

DATA 101


Entity Relationship Diagram (ERD)
Attribute
Name
Prime Attribute
SSN
Student
Relationship
Joins
Attribute
Title
Course
Prime Attribute
CID
Student Entity
Email
Attribute
Duration
Attribute
Course Entity
www.learncomputerscienceonline.com


mongoDB®

DATA 101 Team (Fall 2024)



Lisa Yan
Instructor



Michael Ball
Instructor



Jonathan Ferrari
TA



Rebecca Dang
TA



Christy Quang
TA

CC BY NC SA

File Edit View Run Kernel Git Tabs Settings Help

Terminal 1 | proj0.ipynb

Download Format SQL git Code

Question 2 -- Recent Hits

Select all song titles and artists on the Billboard Hot 100 chart in November 2023 which have ranking of 5 or better, sorted alphabetically by artist name.

Notes:

- The column `week_ending` is a **string**. it is represented `YYYY-MM-DD`. Think back to string comparisons in Data 100. (Later in this class we will explore Postgres date types.)
- Rank 1 is the best, so make sure you are using the right comparison operator!
- Remember that a unique pair of song and artist can appear multiple times on the chart (for example, if the song charts for multiple weeks). Make sure your output only includes each song once.

[23]:
```
result_2 = grading_util.run_file("2")
result_2
```

[23]:

	title	artist
0	Redrum	21 Savage
1	We Can't Be Friends (Wait For Your Love)	Ariana Grande
2	Yes, And?	Ariana Grande
3	Beautiful Things	Benson Boone
4	Texas Hold 'Em	Beyonce
5	Lunch	Billie Eilish
6	Jingle Bell Rock	Bobby Helms
7	Rockin' Around The Christmas Tree	Brenda Lee

2.sql

```
SELECT DISTINCT
    title,
    artist
FROM hot_100
WHERE
    week_ending >= '2023-11-01' AND
    week_ending <= '2024-11-30' AND
    rank <= 5
ORDER BY artist;

```

Simple 1 1 SQL main Mem: 207.87 / 1024.00 MB Ln 1, Col 1 Spaces: 4 2.sql 0

CC BY NC SA

Assignment Workflow

Teacher Setup



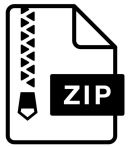
Source notebook



Compiled with Otter



Student notebook





Gradescope autograder

New



`queries` directory with solution `.sql` files



Compiled to skeleton code with script



`queries` directory with skeleton `.sql` files

CC BY NC SA

Student Setup



JupyterHub instance comes preinstalled with dependencies



Student accesses JupyterHub from browser



Gets starter code notebook through `nbgitpuller`

CC BY NC SA

Student Work Cycle

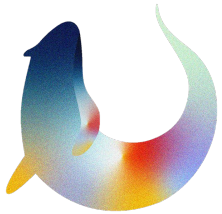
```
SELECT DISTINCT title, artist
FROM hot_100
WHERE artist = 'Taylor Swift'
LIMIT 4;
```

Student writes query



Student thinks about problem

❌ ***Test(s) failed***



✅ ***All tests pass***



Otter creates submission.zip

gradescope by Turnitin

Autograder (with hidden tests) runs when student submits

	title	artist
0	'Tis The Damn Season	Taylor Swift
1	...Ready For It?	Taylor Swift
2	22	Taylor Swift
3	22 (Taylor's Version)	Taylor Swift

Student executes query and views result

Student runs (public) autograder tests and checks correctness

q8d (1.2/2)

```
q8d results:
    q8d - 1 result:
        ✅ Test case passed

    q8d - 2 result:
        ✅ Test case passed

    q8d - 3 result:
        ✅ Test case passed

    q8d - 4 result:
        ❌ Test case failed
        Trying:
            'D' in q8d
        Expecting:
            True
        **********************************************************************
        Line 1, in q8d 3
        Failed example:
            'D' in q8d
        Expected:
            True
        Got:
            False
```

Project 2

● Graded

11 Hours, 19 Minutes Late

Student

[redacted]

Total Points

40.2 / 41 pts

Autograder Score

40.2 / 41.0

Failed Tests

q8d (1.2/2)

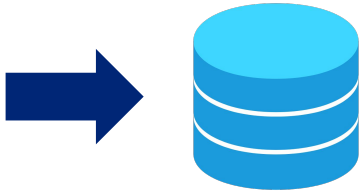
Passed Tests

Public Tests
q0 (1/1)
q1a (1/1)
q1bi (1/1)
q1bii (2/2)
q2a (1/1)
q2bi (1/1)
q2bii (1/1)

Gradescope autograder feedback

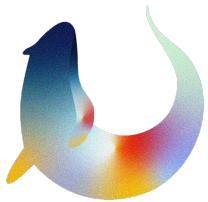
Change in Grading Workflow

```
SELECT DISTINCT title, artist
FROM hot_100
WHERE artist = 'Taylor Swift'
LIMIT 10;
```



	title	artist
0	'Tis The Damn Season	Taylor Swift
1	...Ready For It?	Taylor Swift
2	22	Taylor Swift
3	22 (Taylor's Version)	Taylor Swift
4	Afterglow	Taylor Swift
5	All Of The Girls You Loved Before	Taylor Swift
6	All Too Well	Taylor Swift
7	All Too Well (Taylor's Version)	Taylor Swift
8	All You Had To Do Was Stay (Taylor's Version)	Taylor Swift
9	Anti-Hero	Taylor Swift

Otter autogrades result





stored



`queries` directory stores `.sql` files

New

Old Setup

Pros

- Students write queries directly in notebook

Cons

- Can't provide skeleton code
  - Students forget to wrap queries in `CREATE VIEW`, `WITH … AS`
  - Students create a view but forget to select from it
  - Students forget `LIMIT` → DataHub crashes
  - Hard to reset database by `DROP`-ing views or indexes since each student names theirs differently
- No easy way to reuse queries, e.g. for `EXPLAIN ANALYZE`
- Writing tests was cumbersome
- Saved table results in .pkl file and graded that instead of the query
  - Harder for staff to check for academic misconduct

CC BY NC SA

New Setup

Pros

- Can provide skeleton code
- Can easily reuse queries, e.g. for `EXPLAIN ANALYZE`
- No need to read in .pkl files for unit tests anymore
- We grade students' .sql files
  - Easier for staff to check for academic misconduct
- Grading util that runs the queries uses newer version of psycopg package
- Can run .sql files in command line for quick demoing

Cons

- Students have to switch back and forth between .ipynb and .sql files

CC BY NC SA



Demo

If QR code doesn't work:
tinyurl.com/jc25-data-eng

Thank you!

Link contains **free and open source** code, slides, and contact info

CC BY NC SA