pandas or SQL: which one first?

D8A.ACADEMY/RESOURCES/PANDAS-VS-SQL

01 Same question, both tools

Average basket per city. Neither answer is wrong.

SQL

```sql
SELECT city,
       AVG(total) AS avg_basket
FROM orders
GROUP BY city
ORDER BY 2 DESC;
```

PANDAS

```python
(df.groupby("city")["total"]
   .mean()
   .sort_values(
     ascending=False))
```

02 What SQL is unbeatable at

It runs where the data lives, on data bigger than your laptop.

`filtering early`	WHERE runs on the server. 100M rows never reach you.
`joins`	The database has indexes and a query planner. You do not.
`shared logic`	A view is the same answer for everyone, not one notebook.
`every data job`	SQL is named in 35 of the 89 live postings in our app.

03 What pandas is unbeatable at

Once the data fits in memory, iteration speed wins.

`reshaping`	pivot, melt, stack: painful in SQL, one line here.
`quick charts`	.plot() straight from the frame while exploring.
`ML handoff`	scikit-learn eats DataFrames. There is no SQL API.
`messy files`	CSVs, Excel exports, JSON: pandas reads them all.

04 The pattern: use both

SQL shrinks the data, pandas finishes the job. This is the real workflow.

```python
q = ("SELECT city, total, created_at "
     "FROM orders "
     "WHERE created_at >= '2026-01-01'")
df = pd.read_sql(q, conn)
df.pivot_table(index="city",
  columns=df.created_at.dt.month,
  values="total", aggfunc="mean")
```

→ Push the filter and the join to SQL. Pull only what pandas actually needs.

⚠ COMMON TRAPS

05 Trap: learning both at once

The order matters more than the tools.

`1. SQL first`	Every path here starts there. So do the job postings.
`2. then pandas`	The groupby mental model transfers in a week.
`not in parallel`	Same concepts, different syntax: you will mix them up.

→ SELECT then WHERE then GROUP BY is the same idea as filter then groupby. Learn it once, in SQL.