

Facilitator guide: Module 2, data science and SQL

A teacher-facing companion to Module 2. The module is the full lesson; this guide adds the facilitation layer: a preparation checklist, timing cues, guidance for the discussion questions, and notes on what to emphasize and where students get stuck. Slides and a printable version are below.

At a glance

- **Length:** 90 minutes. This is the longest session in the sequence, since it carries three guided queries plus independent practice.
- **Level:** undergraduate students who have completed an introductory SQL unit, or graduate students in a data analytics or business intelligence program. Comfort with SELECT, WHERE, GROUP BY, ORDER BY, and basic aggregate functions is assumed.
- **Access:** the no-code sandbox at sandbox.altfndata.com, self-registered and auto-approved. An optional code extension shows the same query issued against the production API with a Python client, for students who requested a class API key in advance from info@altfndata.com.
- **Goal of the session:** students leave able to write a filtered, grouped, and ordered query against a real multi-million-row dataset, build a quarterly demand index, compute a pricing power ratio, and explain a data quality issue (unsold lots) they must not discard.

Before class

- Register your own sandbox account and run all three guided queries end to end against the jewelry table, so you have known-good results and row counts on screen.
- If you plan to run the optional API extension, request a class API key from info@altfndata.com well ahead of time and test the Python client (`altfndata_client.py`) or the tutorials notebook (`altfndata_tutorials.ipynb`) once yourself.
- Open the coverage browser and data dictionary tabs beforehand and confirm the documented fields you will reference (`designer`, `model`, `item_title`, `sale_date`, `usd_price_decimal`, `sale_estimates_high_usd_price`, `status`, `vendor`, `stock_ticker`).
- Load the slides and confirm the SQL editor and the pre-built charts tab are both readable from the back of the room.
- Have students register their sandbox accounts before class if you can, to save setup time inside the 90 minutes.

Timed agenda with cues

Time	Segment	What to do	Watch for
0 to 10	Recap and dataset shape	Recap SELECT, WHERE, GROUP BY, ORDER BY, then introduce which fields are documented in this dataset.	Students who recall generic SQL syntax but have never queried a table this large or this messy. Set expectations that some rows will be incomplete.

Time	Segment	What to do	Watch for
10 to 25	Guided query 1	Run the filter-and-sort query live, filtering a single brand's sold lots and ordering by sale_date.	Students missing the % wildcards in the LIKE filter and quietly working with zero or very few rows.
25 to 45	Guided query 2	Build the quarterly demand index with GROUP BY and DATE_TRUNC, then chart it.	Confusion between grouping by quarter alone versus by brand and quarter together when more than one brand is in play.
45 to 60	Guided query 3	Compute pricing power as a ratio of two APPROX_PERCENTILE values and discuss sell-through in preview.	Students dividing by a null or zero high estimate without a guard clause.
60 to 75	Independent practice	Students adapt the demand index query to a brand of their choice and compare results.	Pairs picking a brand with very few sold lots; nudge them to check the row count before trusting the chart.
75 to 85	Optional extension demo	Show the same demand index query issued via the production API with the Python client, projected only.	Do not let this become a live coding exercise; it is a demonstration that the sandbox and the API return the same data.
85 to 90	Wrap-up and homework	Restate the takeaway and hand out the homework.	Leave enough time to state the two-brand, four-query deliverable clearly rather than rushing it at the bell.

Guidance for the discussion questions

Talking points and what to listen for, not a graded key.

- 1. Why filter on status equals sold before computing pricing power.** Listen for: pricing power is a ratio of what was actually paid to what was pre-sale estimated, so unsold lots have no realized price to include; without the filter the ratio would be computed on a mix of real transactions and non-transactions, or fail outright on nulls.
- 2. What a pricing power ratio above 1.0 tells you, and a business reason for conservative estimates.** Listen for: buyers paid over the auction house's forecast, which can reflect genuine demand strength or a deliberately conservative estimate set to attract more bidders into a competitive process.
- 3. Why quarterly bucketing is reasonable here.** Listen for: monthly bucketing is noisy given uneven day-to-day auction calendars, and annual bucketing hides the within-year swings; quarterly is the practical middle ground for a dataset with this transaction cadence.
- 4. Tradeoffs of AVG versus a percentile measure.** Listen for: AVG is pulled by a handful of extreme prices, while APPROX_PERCENTILE (the median) reflects what a typical buyer paid and is far more robust to one exceptional lot.
- 5. Deciding which of two equal-average brands has stronger evidence.** Listen for: lot count matters as much as the average itself; a brand with many more sold lots backing the same average price carries more statistical weight than one resting on a handful of transactions.
- 6. Sandbox versus production API tradeoffs.** Listen for: the sandbox is faster for exploration and one-off analysis, while the API suits automated, repeated, or programmatic pulls that need to be integrated into a larger workflow or model.

7. **Data quality checks before trusting an aggregate.** Listen for: checking row counts per group, confirming the designer filter matched the intended brand, watching for null or zero estimate values, and treating the most recent quarter with caution given ongoing ingestion.

Teaching notes

- **Most common misconception:** students divide `APPROX_PERCENTILE(price)` by `APPROX_PERCENTILE(estimate)` without a guard and are surprised by a null or wildly large ratio. Show the guarded, row-level version of pricing power from the module's Going deeper section and explain why `sale_estimates_high_usd_price > 0` belongs in the `WHERE` clause.
- **Second misconception:** treating a dip or spike in the final one or two quarters of a demand index as a real market trend. Point out plainly that recent periods are still receiving records, so the tail of the series reflects ingestion timing rather than the market.
- **75 minute variant:** drop the optional API extension entirely and shorten independent practice to 10 minutes; keep all three guided queries intact, since building the demand index and the pricing power ratio in sequence is the load-bearing arc of the session.
- **120 minute variant:** after independent practice, have each pair present their chosen brand's demand index to the class, then run the two-brand side-by-side comparison query from Going deeper together to preview Module 3's sell-through concept.
- **If the sandbox is slow or blocked on the room network:** fall back to the pre-run screenshots and exported CSVs you captured in preparation, and assign the guided queries themselves as homework alongside the regular assignment.

For the full lesson content, queries, and homework, see Module 2. Questions or a class API key: info@altfndata.com.