

E4**Row Model**

This FAQ addresses common questions and clarifications that arise while working through the E4 — Row Model exercise. It covers building a row-based P&L model using the Row Model builder — defining Data Source, Formula, and Aggregate node types for Net Profit, Gross Profit, Net Revenue, COGS, and Operating Expenses.

1. Why is Rows configured as Account on the planning sheet, instead of Region Name → Country Name, the way earlier exercises were set up?

Row Model builds its hierarchy on top of whatever rows are on the sheet, so Account is used here because the P&L structure itself — Net Profit, Gross Profit, COGS, and so on — is what needs to appear as rows. Region and Country aren't part of this model's hierarchy; the geographic breakdown isn't the focus of this exercise.

2. Why does the Row Model require deleting all the rows except the root node before building anything?

The default view lists every row as a flat, unstructured list under the All — Root Node. Selecting all rows, deselecting All — Root Node, and clicking Delete clears out everything except that root, leaving a single clean starting point. This way, the hierarchy that follows is built entirely by design, rather than inheriting the default flat structure.

3. What happens to the sheet's original row configuration once the Row Model is enabled and the default row is deleted?

The default Account-based rows are replaced by the driver tree being built — starting from a blank root and rebuilt node by node, rather than inheriting the flat account list that was configured when the sheet was created.

4. What is the difference between a Data Source node and a Formula node?

A Data Source node pulls its value directly from an underlying measure — such as Gross Revenue or Brewing Materials — and is a leaf that reflects actual data. A Formula node calculates its value from other nodes in the tree — such as Net Profit = Gross Profit – Operating Expenses — and has no data of its own, only a calculation referencing its children or other nodes.

5. What is the practical difference between an Aggregate node and a Formula

node, given both COGS and Gross Profit sit at similar levels in the tree?

An Aggregate node like COGS sums its children automatically — Brewing Materials + Packaging + Plant Overhead + Water & Utilities — with no formula to configure. A Formula node like Gross Profit requires an explicit expression — Net Revenue – COGS — because it isn't a simple sum of its children. It's a calculation involving a sibling relationship, not a parent-child rollup.

6. Why is Net Profit configured as a Formula node while Gross Revenue is a Data Source node?

Net Profit is a derived figure that only makes sense as a calculation of other P&L components, so it is built as a Formula. Gross Revenue is a base figure that exists independently in the underlying data, so it is linked directly as a Data Source.

7. What does selecting a source row actually do when configuring a Data Source node like Gross Revenue?

It links that node directly to a specific account in the underlying P&L Data table, so its value is pulled straight from that account rather than calculated. Searching for and selecting Gross Revenue as the source row is what makes this node display the actual Gross Revenue figures from the data.

8. What is the functional difference between Add Child and Add Sibling when building the tree?

Add Child creates a node nested one level below the selected node — a component that rolls into it. Add Sibling creates a node at the same level as the selected node under the same parent — used here to add Returns & Breakage, Distribution Allowance & Rebates, and Federal & State Excise Taxes alongside Gross Revenue under Net Revenue.

9. Does an Aggregate node need its formula configured manually the way Net Revenue, Gross Profit, and Net Profit do?

No — that is the distinction of choosing Aggregate as the node type. It sums its children automatically once they are added, so COGS and Operating Expenses never go through a Configure Formula step, unlike Net Revenue, Gross Profit, and Net Profit, which are explicitly built as Formula nodes.

10. Does the order matter — building COGS before or after Operating Expenses, for instance?

The branches themselves can be built in a different order since COGS and Operating Expenses don't depend on each other. What does matter is that a parent node exists before its children can be added, and that Gross Profit's formula cannot be configured until both Net Revenue and COGS exist as its siblings.

11. Does deleting a node from the Row Model also delete the underlying measure it was linked to?

No — deleting a node only removes it from the tree structure. The underlying data source measure in the P&L Data table is unaffected and remains available to be linked to another node later.

12. If a new GL account needs to be added to COGS later, does the entire Row Model need to be rebuilt?

No — a new Data Source child node can be added under COGS without rebuilding the entire model. Since COGS is an Aggregate node, it will automatically include the new child in its total without any formula changes.

13. Can a Formula node like Net Revenue have its formula edited later, after Gross Profit's formula has already been configured on top of it?

Yes — since Gross Profit's formula references Net Revenue as a node rather than a fixed value, any later change to how Net Revenue is calculated would flow upward and recalculate Gross Profit and Net Profit automatically.

14. What aggregation methods are available for an Aggregate node?

Aggregate nodes support operations including addition, subtraction, multiplication, division, maximum, minimum, first, last, and average.

15. What is the purpose of the other Configure As options (Data Input, Distribution, Driver Input)?

Each option determines how values are populated for that row, based on the kind of input or calculation needed:

-
- **Data Input** – Allows values to be entered directly, with three modes: Static (a fixed value), Rows (input varies by row), and Formulas (input is derived using a formula).
 - **Distribution** – Used when a total value needs to be spread across periods or dimensions, with options to distribute it Equally, by Weights, or by Copy with Trend (based on either value or percentage).
 - **Driver Input** – Used when a row's value needs to be calculated by applying a mathematical operation against another row's value.

16. What is the purpose of the Open Period and Close Period configuration in the Row Model?

This configuration comes into play when a forecast measure is used on a row. It lets you define different behavior depending on the period's status — the Open Period configuration applies to periods still open for forecasting, while the Close Period configuration applies to periods that have already closed (typically once actuals are available). This allows the same row to use different logic depending on whether it's forecasting forward or reflecting finalized data.