

Excel Prompt System to build robust Spreadsheets.

PURPOSE

Explain the context around the spreadsheet task.

- What is currently not working or unreliable now ?
- What problems have you had in the past?
- and importantly, what does success look like?

STEP-BACK

Explain the real workflow and the data model before asking for formulas. This stops the AI guessing incorrectly.

- Business Rules/Logic
- Data grain → what 1 row represents
- Key fields → unique IDs and joins
- Known messy data risks (duplicates, blanks, text numbers)

Existing Workbook Context

Describe

- Sheet names & Column headers.
- Notes about blanks, duplicates, or manual inputs
- AI performs far better when column names are explicit.

Constraints

These are explicit rules the output must follow. They minimize hallucination and misalignment.

- Excel version (365 / 2019)
- Tables allowed or Lists Only.
- Functions allowed (FILTER, XLOOKUP, etc.)
- Must be easy to audit.

Tasks(Scaffold)

This defines what actions AI should perform. They need to be sequential instructions.

- Convert data to tables.
- Add helper columns.
- Create formulas.
- Create summaries / outputs

Verify/Checks

Build Checks for the solution.

- Cross Footing Totals.
- Show Small Test Data & Expected Output.
- Give 3 common Failure cases.

Output Format

Tell the AI exactly what the result must look like.

- Summary of logic.
- Table layout (current vs recommended)
- Where each formula goes.

The Prompt Works as it narrows the Probability Space Step by Step.

- Purpose → Why
- Existing Context → What exists
- Step-Back → Understand the data
- Constraints → What's allowed
- Task → How to solve
- Verify → How to trust it
- Output → What to deliver

3. Let AI Build the Prompt With You.

You don't need to write the full prompt yourself. Download for free our full prompt text template here at [OnlineExcelTraining.com](https://www.onlineexceltraining.com)

Step 1: Fill in the parts in red.

Fill in the human contact part of the template : the real-world goal, existing problems, Excel version, existing sheets & tables and what changes are allowed. (helper columns, new sheet, restructure).

Step 2: AI completes the Prompt & Solve the Task.

Using this instruction: "Using the Excel prompt template, complete the prompt, list assumptions and risks, then solve the task."

AI will add:

- assumptions and risks.
- data structure thinking.
- verification checks.
- constraints and output structure.

Step 3 — Implement and Verify.

Paste Solution into Excel and run the Verification Checklist. Test and retest the solution.

Real Life Prompt Examples

1. COMPARING TWO LISTS

Problem: Find staff who have NOT done training by comparing Two Lists.

PURPOSE: Find staff who exist in the Main Staff List but do NOT appear in the Training list. **Success looks like:** list of staff who have not done training yet.

EXISTING WORKBOOK CONTEXT: Sheet: StaffList (Main list)

Columns: StaffID | Name | Email.

Sheet: TrainingCompleted. Columns: StaffID | Name | Email | TrainingDate

STEP BACK: Data Grain.

- StaffList → 1 row = 1 employee (master)
- TrainingCompleted → 1 row = 1 training record : StaffID = unique key
- Constraints: (Excel 365, scalable), Excel Tables are allowed. UNIQUE, FILTER, XLOOKUP are allowed.

Task: Convert lists into Tables named: StaffList & TrainingCompleted. On a new sheet called ReminderList, generate a dynamic list of staff who:

Exist in StaffList but Do NOT appear in TrainingCompleted

Return the following columns in the ReminderList:

- StaffID: Name: Email

Verify: Total Staff = rows in StaffList. Staff needing reminder + completed training = total staff

OUTPUT EXPECTED: A dynamic table showing: Staff who still need to complete training. • Email-ready list (StaffID, Name, Email), Sorted alphabetically, Verification checks confirming totals match

2. FEES SUMMARY SHEET

Problem: Calculate royalties due to thousands of artists.

PURPOSE: I need to calculate royalties per unique artist where \$2 is paid per song played. There are over 100,000 records and the same artists appear many times.

Success Looks Like: artist gets \$2 every time their song is played but duplicates in the Singer column could lead to double counting.

STEP BACK: Data Grain.

- StaffList → 1 row = 1 employee (master)
- TrainingCompleted → 1 row = 1 training record : StaffID = key

EXISTING WORKBOOK CONTEXT:

Sheet: SongLog Columns: Date | Song | Singer | DurationSeconds

Constraints: (Excel 365, scalable). Prefer modern dynamic formulas. Solution must scale efficiently to 100k+ rows

Task: Convert the SongLog range into an Excel Table named SongLog. Create a helper column to clean Singer names. Generate a unique list of artists using dynamic array functions. Count number of plays per artist using structured references. Calculate royalties owed per artist at \$2 per play. Place results in summary sheet.

Verify: Total plays in summary = number of rows in SongLog and Total royalties in summary = total royalties from SongLog

OUTPUT EXPECTED: A dynamic summary table showing: Artist name, Number of plays, Royalties owed (currency formatted), Sorted highest to lowest royalties and verification checks confirming totals match.

Follow-up Questions. AI work is often Iterative.

- What edge cases could break this spreadsheet and how can we protect against them?
- How would you redesign this if the dataset grows to 100k rows?
- How can we turn this into a reusable template for future projects?
- How can we make this spreadsheet safer so users can't accidentally break it?
- How could we add a simple dashboard or summary view on top of this?
- Can you write short instructions to include in the workbook explaining how it works?
- What checks can we add to confirm the results remain correct each month?
- Can you explain the logic behind each helper column in plain English?
- How can we restructure this to use Excel Tables and structured references?

Common AI Errors & the Excel Skills Required to Spot and Fix Them.

Cell reference types (CRITICAL):

Users must understand: Relative & Absolute reference → \$A\$2 and Mixed reference → \$A2 or A\$2. AI often gives a formula that works in row 2 but breaks when copied down. This is one of the biggest real-world spreadsheet risks.

Sheet linking syntax:

Users should recognise: `Sheet1!A2` : 'Sales Data'!B:B
AI sometimes: References the wrong sheet: Forgets quotes for spaces :Mixes sheet names. A single wrong sheet reference can corrupt the whole model.

Table vs Cell references:

Users should understand the difference between a Cell reference: `SUMIFS(B:B, A:A, E2)` and a Table reference: `SUMIFS(tblSales[Amount], tblSales[Customer], E2)`
AI often mixes them incorrectly or switches mid-formula. Tables are usually safer and easier to audit.

Column alignment in SUMIFS / COUNTIFS.

Users must know the rule: Every range must be the same size and shape. Example error AI sometimes makes: `SUMIFS(B:B, A:A, E2, C2:C500, F2)` That formula will run — but may be wrong.

Structured formula placement.

Users should check which column the formula belongs in. What the starting row is or how it should be copied. AI sometimes writes correct formulas but gives vague placement. Placement mistakes = broken results.

Whole column references vs ranges.

AI often uses: `A:A, B:B` Users should know: This can slow large files: It can also include unintended rows: Tables or defined ranges are safer.

Data Structure Errors (very common).

- Does one row represent one thing consistently?
 - Are multiple data "levels" mixed in one table.
- Warning signs: Totals seem too high/low: Duplicate counting: Missing records:

Duplicate & Missing Data Checks.

- Are key columns unique where they should be?
 - Are any records missing from summaries?
- Quick checks: COUNT rows vs COUNT of unique IDs: Compare totals before vs after processing.

Edge Case Testing.

- What happens if a cell is blank?
- What happens if there are no matches?
- What happens with a new row of data?

