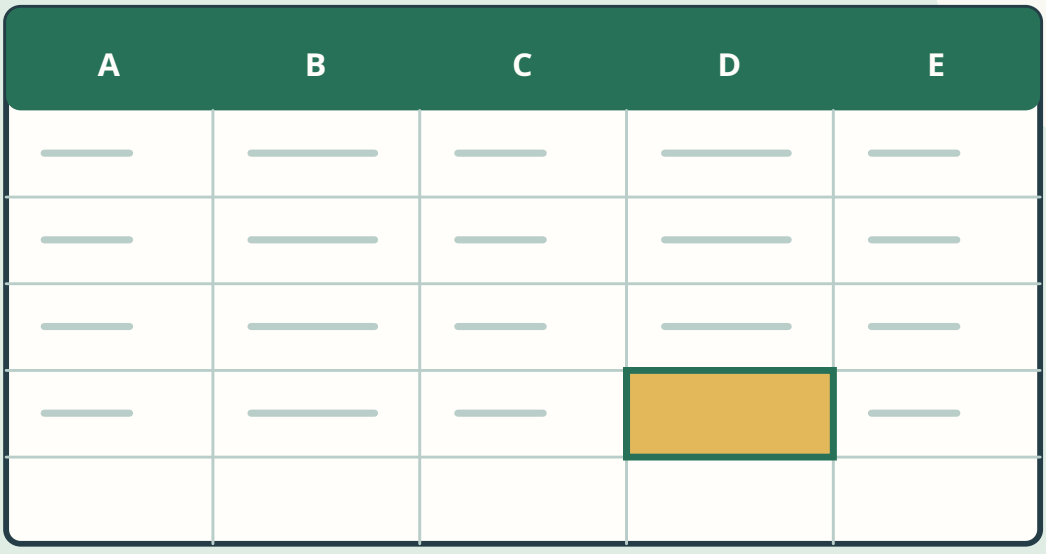
VISIBLE FORMULA ERRORS

Handling Formula Errors Without Hiding Problems



ERROR / CAUSE

Handling Formula Errors Without Hiding Problems

Learning outcomes

- Diagnose six planted spreadsheet errors by cause.
- Repair structural faults rather than hiding them.
- Use visible markers for missing denominators and unregistered keys.
- Distinguish true blank, formula-generated empty text, whitespace, numeric zero, and numeric text using explicit tests.
- Trace a propagated error to its first failing precedent and repair only from a verified cell specification or correction ticket.
- Build type- and missingness-aware ratio guards while preserving genuine zero and exposing unexpected structural faults.
- Diagnose lookup no-match, missing key, duplicate identity, missing return field, and dirty-key cases without inventing registry data.
- Use explicit separator conventions for approved numeric-text conversion while preserving identifiers and raw evidence.
- Separate human-readable status from unavailable numeric results and report coverage alongside partial summaries.
- Design regression tests for valid, missing, zero, text, boundary, and structural-fault cases, with temporary inputs restored.
- Produce a reproducible eight-record correction and exception package with evidence-linked repairs and an honest release decision.

Audience: General workplace learners using synthetic practice materials

Setup

Use only the supplied fictional records. Do not access real accounts, visit practice URLs, change live permissions, execute downloads, or submit messages. For spreadsheet exercises use a blank local practice workbook with English Excel formula names and comma separators; some regional installations use semicolons. Keep raw inputs and worked outputs separate.

Study the foundations, work through the applied units, complete the practice tasks, and then attempt the assessment. Use only the supplied fictional practice data. Keep your original inputs and your revised work separate.

Core concepts

Separate causes before choosing a handler

A formula error is information about a failed calculation. #REF! points to an invalid reference, #DIV/0! to a zero or empty divisor in ordinary division, #N/A often to a missing lookup match, #VALUE! to an incompatible value, and #NAME? to an unrecognized name such as a misspelled function. The displayed code is a starting clue, not the full diagnosis. Inspect the formula and its inputs. A wrong but numeric result can also be a defect, so absence of error codes is not proof of correctness.

Repair a broken model at its source

If a formula contains #REF! after a referenced cell or range was removed, determine the intended input and reconnect it. Do not wrap the broken reference in IFERROR and label the resulting zero as a valid quantity. Likewise, correct a misspelled function rather than hiding #NAME?. If a quantity is the word ten but the verified practice source says numeric 10, correct that source input and keep a note. Never guess a replacement value when the authoritative source is missing.

Make expected missingness explicit

An absent lookup key can be an expected exception and should show UNREGISTERED. A missing denominator should show Missing denominator, while an actual numeric zero denominator should show Zero denominator. These are different reasons the ratio is unavailable. A guard such as =IF(C2="","Missing denominator",IF(C2=0,"Zero denominator",B2/C2)) preserves that distinction for the supplied numeric cases. Do not report an unavailable ratio as zero, because a genuine ratio of zero has a different meaning.

Use error handling narrowly and test recovery

IFERROR returns a replacement for many error types, so it can hide unrelated faults if applied indiscriminately. Prefer an explicit input test or a lookup’s not-found argument when the expected exception is known. Keep unexpected structural errors visible for repair. After correcting the cause, verify an independent expected result and test a valid replacement input. An exception log should record case, observed error, cause, repair or warning, and verification. The aim is understandable behavior, not a sheet that merely contains no visible error symbols.

Worked example

A ratio is =B2/C2, with B2=12 and C2 blank. Replacing it with =IFERROR(B2/C2,0) would make the result look like a genuine zero ratio and conceal that input is missing. Instead use =IF(C2="","Missing denominator",IF(C2=0,"Zero denominator",B2/C2)). The initial result is Missing denominator. Entering C2=3 in a temporary test produces 4. Entering C2=0 produces Zero denominator. Restore the blank afterward. This tests both the exception labels and the valid numerical path.

Foundation practice

Case workbooks

Use six separate practice sheets named E1 through E6 so cell addresses below do not conflict. Each case is self-contained; no live data is used.

Six cases

- E1: B2=8; H2=0.5; broken D2 formula =B2*#REF!. Verified specification: multiply quantity B2 by shared rate H2.
- E2: B2=12; C2 genuinely blank; D2 =B2/C2. Specification: ratio unavailable until denominator is supplied.
- E3: B2=12; C2=0; D2 =B2/C2. Specification: zero denominator is invalid for this ratio, not a zero result.
- E4: A2="DOC-99"; F2="DOC-01",G2="Shelf-A"; F3="DOC-02",G3="Shelf-B"; B2 =XLOOKUP(A2,F2:F3,G2:G3). DOC-99 is unregistered.
- E5: B2="ten"; C2=3; D2 =B2*C2. Verified source ticket explicitly says quantity numeric 10, not the text ten.
- E6: B2=2; B3=3; B4=4; D2 =SUMM(B2:B4). Specification: sum the three supplied numbers.

Required output

A six-case correction sheet or ledger with cause, corrected formula/input, visible exception message where needed, and independent check.

Foundation tasks

- Classify and repair all six cases without using one blanket blank-output IFERROR formula.
- For E2 and E3, preserve different warnings and test a valid denominator of 3 in a copy.
- Capstone: explain why a sheet whose errors have all been replaced with zero can understate totals or make unavailable metrics appear valid.

01 Define what a result is allowed to mean

A repair starts with a statement of meaning, not a replacement formula. In this course, fictional Northfield Services records completed items and available work slots. A ratio means completed items divided by available slots for the same record. A numeric zero means the record was measured, the denominator was positive, and no items were completed. An unavailable result means the evidence does not yet support that calculation.

Use assets/README.txt to construct the practice sheets in a new local workbook. Import each CSV with its first row at A1 and retain a separate unmodified copy of the files. CSV files carry values, not workbook formulas, formatting, or reliable cell types. The cell map therefore identifies deliberately blank cells and text entries that need special handling. A printed instruction such as [blank] describes an empty cell; do not type those brackets into the

workbook. Enter formulas only from the supplied cell-map instructions, never from an unexpected source record.

Three kinds of evidence should remain distinguishable. Raw evidence is the original supplied value, such as the word twelve. Prepared evidence is a verified numeric replacement in a working copy. Calculated evidence is a formula result whose inputs and references can be inspected. Overwriting the raw file collapses these layers and makes it difficult to explain whether a later improvement came from better data or a changed calculation.

The RatioLab dataset contains eight records, R01 through R08, in rows 2 through 9. Column B is Completed and column C is Slots. Its fictional rule permits nonnegative numeric counts, requires a positive numeric denominator, and rejects a completed count greater than slots. Not all lessons use this final validation rule immediately; early units deliberately isolate one defect at a time. Do not silently apply a more permissive rule merely because Excel can calculate a number.

Record the application version used for the practice. Core IF, ISNUMBER, ISBLANK, COUNTIF, SUM, and AVERAGE examples do not require dynamic arrays. XLOOKUP examples require a supporting edition such as Microsoft 365, Excel 2021, or Excel 2024; Excel 2016 and 2019 need the supplied exact-match VLOOKUP alternative. English function names and comma argument separators are used throughout. Regional installations may require translated function names or semicolon separators; that is a syntax adaptation, not a change to the business rule.

Case excerpt: R02 in row 3 has B3=0 and C3=5; R03 in row 4 has B4=18 and C4 blank; R07 in row 8 has B8 blank and C8=4. The complete eight-record source is assets/ratio_lab.csv.

E01-X1 Define what a result is allowed to mean — application

Using the definitions above, classify R02, R03, and R07 as a measured zero, missing denominator, or missing numerator. State whether replacing each unavailable result with 0 would preserve its meaning. Do not yet apply the completed-greater-than-slots rule.

Submit: Three classification records containing ID, input evidence, allowed result meaning, and the reason a substitute zero is or is not justified.

02 Classify symptoms before diagnosing causes

An error code identifies a failed operation, but it rarely supplies the complete cause. If a denominator contains a formula that already returns #N/A, dividing by that cell can propagate #N/A even though the visible formula uses division. Conversely, a denominator that is genuinely empty can cause #DIV/0!. Diagnosis therefore requires both the displayed symptom and inspection of the cells that feed it. Do not assign a repair solely by matching a code to a memorized phrase.

Separate three families of defect. Calculation errors are values such as #REF! or #VALUE! that Excel can display. Data exceptions are conditions that the model expects to encounter, such as a requested code absent from a reference list. Semantic errors are results that calculate successfully but contradict the specification, such as multiplying by the adjacent

record's rate. These families can overlap. A missing registered rate might first appear as a data exception, then propagate into a calculation error downstream.

For the miniature Diagnostic sheet, put 6 in B2 and 2 in C2. The correct ratio is 3. A formula using B2*C2 returns 12 with no error code; a formula using B2/C2 returns 3. Both are syntactically valid. Only comparison with the ratio specification identifies the first as wrong.

Use a hypothesis-and-evidence note. Write the observed output, the formula, one suspected cause, and the next cell to inspect. Then test the hypothesis in a disposable copy with a known valid input. If changing a denominator from empty to 2 causes 6/2 to become 3, the observation supports a missing-denominator diagnosis. It does not establish that 2 is the correct permanent source value. Testing and authorizing a repair are separate acts.

The table gives starting clues rather than guarantees. A #NAME? may be a misspelled function, an undefined name, or an unsupported function; inspect the actual expression. A #N/A may be deliberately produced with NA() rather than generated by a lookup. Record that distinction because a deliberate unavailable marker is functioning as designed, while an unexpected error needs investigation. Retain the original observation even after the cause is resolved so a reviewer can see what changed.

Observed clue	First inspection	What not to assume
#REF!	Formula references and deleted ranges	That a zero substitute repairs the reference
#DIV/0!	Divisor and its precedents	That the numerator is zero
#VALUE!	Types and operations	That every text input should become zero
#N/A	Lookup match or deliberate NA()	That every case means an unregistered code
#NAME?	Function spelling and names	That hiding it resolves compatibility

E02-X1 Classify symptoms before diagnosing causes — application

On Diagnostic, B2=6, C2=2, D2=B2*C2; B3=6, C3 is blank, D3=B3/C3; B4=6, C4=2, D4=B4/#REF!. Identify the symptom family and the first justified action for each row. The intended operation in all rows is B divided by C.

Submit: Three diagnoses, including the independent expected result where available.

03 Trace the first failing precedent

When several cells show the same error, there may be only one initiating defect. On a sheet named Chain, enter B2=30 and leave C2 genuinely empty. Put =B2/C2 in D2, =D2*2 in E2, and =E2+5 in F2. D2 fails first, and the error flows through E2 and F2. Editing all three cells would create three opportunities for inconsistent logic. The useful repair point is the dependency that lacks a valid denominator, not whichever final cell a reader notices first.

Trace the chain backward from F2. Its own addition is reasonable if E2 is numeric. E2's multiplication is reasonable if D2 is numeric. D2 is the first place where the input rule fails. Write the dependency sequence C2 to D2 to E2 to F2, and keep B2 as the second input to D2. In desktop Excel, formula-auditing features can help expose references; a manual cell-by-cell inspection remains sufficient for this small model and avoids reliance on a particular interface layout.

A missing-data warning belongs near the calculation that understands its meaning. Replacing F2 with =IFERROR(E2+5,0) leaves the upstream defect unresolved and produces a misleading headline. Replacing D2 with a text warning without redesigning E2 also creates a problem: E2 tries to multiply text. The lesson is not that visible text is bad. It is that downstream consumers need a deliberate contract for unavailable values, which later units implement with separate status and numeric channels.

For this diagnostic exercise, keep the original formulas and annotate the missing input rather than disguising the propagated errors. In a test copy only, set C2 to 5. D2 should become 6, E2 should become 12, and F2 should become 17. Verify those steps by arithmetic outside the workbook. Restore C2 to empty afterward.

One successful test establishes only the path exercised. It does not prove the model correctly handles zero, text, negative values, or future row insertions. Save the three expected numerical outputs and the restored exceptional state as separate observations. If the final result still differs, inspect the earliest point of divergence rather than adding another outer error handler.

```
Chain!D2 =B2/C2
Chain!E2 =D2*2
Chain!F2 =E2+5
```

E03-X1 Trace the first failing precedent — application

In a separate Chain row 3, use B3=42, C3 blank, D3=B3/C3, E3=D3*3, and F3=E3-2. Identify the initiating input defect. Test with C3=7, record all three results, then restore the blank. Explain why changing F3 alone is not a source repair.

Submit: A dependency sequence and a before/test/restored log with exact numeric results for the test state.

04 Distinguish blank, empty text, whitespace, and zero

Cells that look empty are not necessarily equivalent. A genuinely unused cell has no stored value or formula. A cell containing =”” has a formula whose result is empty text. A cell containing one space contains text with a character, even when that character is difficult to see. A numeric zero is a measured value. These differences can affect comparisons, counts, lookup keys, and mathematical operations, so the diagnosis should describe the stored state rather than its appearance.

Create BlankLab with the four states shown below. ISBLANK tests whether the referenced cell is actually empty; it does not treat a formula returning empty text as an empty cell. A comparison to an empty string is useful for the narrower practice policy that groups true

blanks and formula-generated empty text as missing. ISNUMBER distinguishes numeric zero from all three nonnumeric states. LEN is useful for exposing the one-character space in this controlled example, but it is not a universal test for valid content.

The business rule must say which states count as missing. For Northfield's imported counts, true blanks and empty text mean not supplied. A space-only entry is an invalid text value that needs normalization or source correction. Numeric zero remains numeric evidence. This choice prevents a whitespace typo from being quietly reclassified as a properly missing record, and prevents a legitimate zero measurement from disappearing into an empty-looking display.

Do not use cell color or alignment as the deciding test. Formatting can make a number look like text or a zero appear blank. A value may also have been pasted into a preformatted cell. Inspect the formula bar and use an explicit test in an adjacent diagnostic cell.

There is also an order-of-tests issue. A true blank can compare equal to zero in some formula contexts. If a guard tests C2=0 before testing C2="", it can call a missing denominator a zero denominator. Check missingness first when those states require different labels. The general lesson is to define categories that do not accidentally overlap, then order the checks so the more specific missing-data interpretation is not swallowed by a broader numeric comparison.

Cell	How to create it	ISBLANK	Cell=""	ISNUMBER	LEN
B2	Clear contents	TRUE	TRUE	FALSE	0
B3	Enter formula =""	FALSE	TRUE	FALSE	0
B4	Type one ordinary space	FALSE	FALSE	FALSE	1
B5	Enter numeric 0	FALSE	FALSE	TRUE	1

E04-X1 Distinguish blank, empty text, whitespace, and zero — application

Create B6 with formula ="" and B7 with numeric 0. Predict ISBLANK, the comparison to "", and ISNUMBER for each. Then explain which denominator warning each should produce under the stated policy. Check the predictions in your workbook.

Submit: Six Boolean predictions and two warning labels, with one sentence explaining test order.

05 Build a guard from a stated input contract

A guard should encode the input rule rather than catch every possible failure after calculation. Begin with a narrow teaching contract: the numerator must be present and numeric; the denominator must be present, numeric, and nonzero. Negative values and ratios greater than one are not yet checked in this unit. This deliberately limited contract lets you inspect the meaning of each branch before adding Northfield's stricter nonnegative-count rule in the capstone.

The formula below tests the numerator first, then the denominator, then performs division. A blank numerator receives its own label instead of being coerced to a plausible zero. Text such as twelve is rejected rather than translated by guesswork. A denominator stored as the text "3" is not accepted by ISNUMBER even if some arithmetic operators would coerce it. Consistent input types are part of the stated contract, not merely an obstacle to making Excel display a result.

The check order is a reporting choice when a row has multiple defects. If both B2 and C2 are blank, this formula reports Missing numerator first. That does not assert the denominator is valid. It identifies the first failed rule. For a compact operational result, document this precedence so another user does not interpret the first warning as a complete inventory of problems.

Apply the formula to the eight RatioLab records without changing their raw inputs. R01 produces 4, R02 produces 0, and R08 produces 3 under this arithmetic-only contract. R03 needs a denominator, R04 has a zero denominator, R05 has a nonnumeric numerator, R06 has a text denominator, and R07 needs a numerator. The later range rule will reject some of the currently numeric ratios; the output here demonstrates type and missingness validation, not final operational approval.

Add a helper status column when several checks must be inspected independently. Also distinguish known input exceptions from structural faults. A broken reference in this formula may still show an error code, which is useful: the guard should not convert an accidental deletion into the reassuring label Missing denominator. You want expected data states explained and unexpected model faults exposed.

```
=IF(B2="","Missing numerator",
 IF(NOT(ISNUMBER(B2)),"Invalid numerator",
 IF(C2="","Missing denominator",
 IF(NOT(ISNUMBER(C2)),"Invalid denominator",
 IF(C2=0,"Zero denominator",B2/C2)))))
```

E05-X1 Build a guard from a stated input contract — application

On GuardTest enter (B2,C2)=(0,4), (B3,C3)=(8,blank), (B4,C4)=(8,0), and (B5,C5)=(8,text "4"). Apply the guard to each row. List the four outputs. State the output if both inputs in row 6 are blank.

Submit: Five exact results, plus a note identifying whether the all-blank row has one or two input defects.

06 Correct numeric text without inventing source values

A numeric-looking string and a word for a number are different repair problems. The text "003" can be a quantity or an identifier; the field definition decides whether converting it to 3 preserves meaning. The word twelve is not an ordinary numeric string. Replacing it with 12 is justified only when a verified source establishes that quantity.

ImportLab supplies four text tokens in column B and declared formats in column C. I01 is "1,250.50" with dot decimal and comma grouping. I02 is "1.250,50" with comma decimal and dot grouping. I03 is twelve and has ticket T101 confirming numeric 12. I04 is "003" and is an identifier, not a quantity. Set column B to Text before entering or importing the tokens. The quotation marks here describe strings; they are not part of the stored values.

NUMBERVALUE permits explicit decimal and grouping separators. In a prepared output cell, =NUMBERVALUE(B2,".",",") converts I01 to 1250.5, while =NUMBERVALUE(B3,",",".") converts I02 to the same number. Do not apply one separator convention to an unclassified mixed-format column. The token "1,250" can mean one thousand two hundred fifty or one and a quarter under different conventions. A successful conversion is not proof that the selected convention was correct.

Missingness should be checked before conversion. NUMBERVALUE can turn empty text into zero, which is not acceptable when the input was not supplied. Use a separate missing-input branch and retain the original token. If conversion fails, a targeted visible parse warning is more honest than a zero substitute. A parsing handler is acceptable only when its scope is explicit and the raw value remains available; it must not wrap a larger expression that also contains lookups and structural dependencies.

For the word twelve, write a correction record linking I03 to T101 and prepare numeric 12. Do not change I04 merely because a number function would accept its digits. Losing leading zeroes can break later exact-match lookups, and that damage can be invisible until records are joined. Recheck both type and meaning after normalization: a numeric quantity should pass ISNUMBER, while an identifier may properly remain text.

Ticket	Record	Verified evidence
T101	ImportLab I03	Confirmed quantity is numeric 12
T102	RatioLab R05	Confirmed completed quantity is numeric 12
T103	RatioLab R06	Confirmed slots are numeric 3
T104	Capstone C06	Confirmed completed quantity is numeric 8

E06-X1 Correct numeric text without inventing source values — application

Prepare results for I01 through I04 using the stated format and ticket. Calculate half of each valid numeric quantity. Do not calculate half of the identifier. Explain why automatically converting every token, including a blank token, is not an acceptable policy.

Submit: A four-row preparation record with raw token, action, prepared value/type, and half-quantity when meaningful.

07 Repair references using the specification

A broken reference is a problem in the model's wiring. If a formula contains #REF!, replacing its output with zero only changes what a reader sees. It does not restore the intended input. The correct replacement reference must come from the workbook specification or another verified record, not from whichever adjacent cell happens to contain a convenient number. When the source cannot be established, retain the defect and describe what evidence is needed.

RatesLab contains quantities 8, 12, and 20 in B2:B4. H1 is Shared rate and H2 is numeric 0.75. The supplied specification says each output equals that row's quantity multiplied by H2. D2 is deliberately broken as =B2*#REF!. Because the specification identifies both operands, the repair is =B2*$H$2. The independent result is 6. The dollar signs hold both the shared rate's column and row fixed when the formula is copied.

Fill the repaired expression down to D4. The expected formulas become =B3*$H$2 and =B4*$H$2, giving 9 and 15. The three outputs sum to 30. Independently, total quantity 40 multiplied by rate 0.75 also equals 30. This reconciliation checks the calculation relationship without simply summing a second copy of the same potentially broken output formulas. Independent checks are strongest when they approach the expected result differently.

Do not assume an absolute reference protects the cell from every structural change. Dollar signs control how a reference adjusts during copying; they are not a guarantee against deleting the referenced cell, replacing its contents, or modifying workbook structure. After any such change, inspect the resulting formula and compare with the specification. The intent to use a shared rate should remain visible in the cell label and the repair record.

The repair should preserve a concise before-and-after record: original formula, observed error, verified intended input, corrected formula, and test result. If someone later asks why H2 rather than H3 was used, the evidence should answer that question. In this exercise, neither hiding the error nor inserting an unexplained constant 0.75 directly into every formula is the best maintainable solution.

```
RatesLab!H2: 0.75
D2 =B2*$H$2
D3 =B3*$H$2
D4 =B4*$H$2
```

E07-X1 Repair references using the specification — application

Extend RatesLab with B5=16. Fill the corrected formula into D5. Calculate the four output values and total. Then predict the new four values and total if the authorized shared rate changes to 0.5. Restore H2 to 0.75 after testing.

Submit: Two four-value result sets, their totals, the exact D5 formula, and restoration evidence.

08 Find wrong numbers that do not trigger an error

Not every reference fault produces #REF!. A reference can point to a real but unintended cell and still return an ordinary number. To find them, compare formula patterns, input relationships, and an independent expected result rather than relying on the absence of warning symbols.

On DriftLab, enter quantities 10, 20, and 30 in B2:B4. Put 0.4 in H2, 0.1 in H3, and 0.9 in H4. Only H2 is the authorized shared rate; H3 and H4 are unrelated scenario notes. Enter =B2*H2 in D2 and copy it down. Excel adjusts the relative references, so D3 uses H3 and D4 uses H4. The outputs become 4, 2, and 27. All are numeric, and their sum 33 may look plausible to someone who does not inspect the rule.

The authorized calculation requires every quantity to use H2. Correct the starting formula to =B2*$H$2 and fill down again. Outputs become 4, 8, and 12, with total 24. The difference between the faulty and correct totals is 9. A total-only check reveals a discrepancy, but a row-level check identifies exactly which rows are affected. Row 2 happens to be right in both versions, so checking only the first row would have missed the defect.

Use the first, middle, and last row to inspect a copied formula family, then check any boundary where the pattern changes. A different-looking formula is not automatically wrong: a subtotal may intentionally differ from detail rows. The pattern comparison is a prompt to consult the specification. Likewise, a uniform pattern can be consistently wrong if every formula uses multiplication where division was intended. Pattern, arithmetic, and meaning must agree.

The best regression test targets the reason for the original defect. Change H3 in a test copy and confirm that the corrected D3 does not change, because H3 is not an authorized input. Change H2 and confirm all three outputs update. These tests demonstrate reference independence and dependence respectively. They are stronger than merely observing the corrected total once.

E08-X1 Find wrong numbers that do not trigger an error — application

After correcting DriftLab, set H3 to 7 in a copy while H2 remains 0.4. Predict D3 and the total. Then restore H3 and set H2 to 0.25. Predict all three outputs and the total. Explain which test would expose a remaining relative-reference defect.

Submit: The two test scenarios, exact results, and a causal explanation.

09 Separate misspellings, undefined names, and compatibility

The #NAME? family deserves inspection rather than a generic error label. A function name may be misspelled, a named range may not exist, or text may be missing quotation marks. An unsupported function can also make a workbook unusable in an older application. These causes require different repairs even though the symptom looks similar.

Create NameLab with B2=4, B3=7, and B4=9. The formula =SUMM(B2:B4) contains an extra M; the intended function is SUM, and the verified result after repair is 20. Next inspect =B2*SharedRate. This is meaningful only if SharedRate is a defined name pointing to the authorized rate. For this exercise no such name exists, while H2 is explicitly specified as the rate 0.25. Replacing the undefined name with $H$2 produces 1 without creating an undocumented naming convention.

Quotation marks also matter. In a formula, the marker UNREGISTERED is text and must be written as "UNREGISTERED". Without quotes, Excel attempts to interpret it as a name. The repair is to quote the intended label, not to create a named range whose value happens to contain that label. This small syntax difference is especially easy to overlook when copying formulas from a document with typographic quotation marks. Use straight double quotation marks in the actual formula.

Compatibility is a separate investigation. XLOOKUP is not available in Excel 2016 or Excel 2019, even if a workbook created elsewhere contains it. For the simple two-column registry in this course, =IFNA(VLOOKUP(A2,Registry!$A$2:$B$5,2,FALSE),"UNREGISTERED") supplies an exact-match alternative. The FALSE argument is essential to this model; an approximate match may return a plausible but wrong location. IFNA limits replacement to #N/A, although later units explain why even that requires care with errors in returned cells.

Document the formula variant used and test one existing key plus one absent key. Do not claim compatibility based only on the presence of a product name in a file. The intended function must be supported in the application that will open the workbook.

```
NameLab!D2 =SUM(B2:B4)
NameLab!D3 =B2*$H$2
Legacy exact lookup:
=IFNA(VLOOKUP(A2,Registry!$A$2:$B$5,2,FALSE),"UNREGISTERED")
```

E09-X1 Separate misspellings, undefined names, and compatibility — application

With H2=0.25, repair =SUMM(B2:B4) and =B3*SharedRate under the supplied specification. State their results. In the legacy lookup, explain why replacing FALSE with TRUE is not an acceptable compatibility fix.

Submit: Two repaired formulas with results and an exact-match justification.

10 Treat no-match, duplicate keys, and blank keys differently

A lookup is a relationship between an input key and an authoritative registry. An absent key means no matching record was found in the specified range. A duplicate key means multiple records could satisfy the identity rule. A blank request means no usable key was supplied.

Registry contains DOC-01 with Shelf-A, DOC-02 with Shelf-B, DOC-03 with an empty location, and DOC-04 with Shelf-D in A2:B5. RequestLab includes DOC-02, DOC-99, a blank key, DOC-01 followed by one trailing space, and DOC-03. With an exact XLOOKUP, DOC-02 returns

Shelf-B and DOC-99 can return the chosen UNREGISTERED marker. Nothing in that behavior authorizes adding DOC-99 to the registry or assigning a guessed shelf.

Before lookup, use a blank-key guard. =IF(A2="","Missing key",XLOOKUP(A2,Registry!$A$2:$A$5,Registry!$B$2:$B$5,"UNREGISTERED",0)) keeps an absent request distinguishable from a supplied but unregistered code. The explicit 0 match mode documents exact matching even though it is the default. The fixed ranges prevent the lookup table from drifting as the request formula is filled downward. Registry quality checks remain necessary because matching a key does not prove its returned field is usable.

To demonstrate duplicates safely, create DuplicateLab with DOC-02 in A2 and A3, Shelf-B in B2, and Shelf-Z in B3. A normal forward XLOOKUP returns the first match, Shelf-B. That is a defined search behavior, not a resolution of the conflicting registry evidence. =COUNTIF(A2:A3,"DOC-02") returns 2 and makes the ambiguity visible. In this controlled dataset, codes contain no wildcard characters, so COUNTIF criteria behave as intended.

The fictional release rule requires exactly one registry entry per nonblank request key. A count of zero supports UNREGISTERED; a count above one requires Duplicate key review. Do not delete the inconvenient duplicate merely to obtain a result. The authorized registry owner would need to establish which record is correct. A reliable workflow can therefore leave a case unresolved while still being complete as a diagnostic task: the learner has identified the obstacle, preserved the evidence, and avoided fabricating a location.

State	Evidence	Required interpretation
Missing key	Request cell empty	No lookup identity supplied
Unregistered	Nonblank key; match count 0	No registry record
Unique	Match count 1	Inspect the matched return field
Duplicate	Match count greater than 1	Identity relationship is ambiguous

E10-X1 Treat no-match, duplicate keys, and blank keys differently — application

Use DuplicateLab and a separate request DOC-99. Give the match counts for these two nonblank keys and the required label for each. For a blank request, state why the blank-key guard applies before counting. Explain why Shelf-B is not a sufficient approved result for the duplicated key.

Submit: Three classification rows and a short explanation of first-match behavior versus data authority.

11 Normalize keys only under an explicit identity rule

An exact-match failure can be caused by a dirty key rather than an absent registry record. RequestLab's fourth request is DOC-01 followed by one ordinary trailing space. That string is not identical to DOC-01. Before concluding that a new code needs registration, inspect the stored characters and compare them with the authoritative format. However, similarity alone does not authorize rewriting identifiers: some systems intentionally distinguish spaces, punctuation, or leading zeroes.

For this practice only, the registry specification says document keys contain uppercase DOC, one hyphen, and two digits, with no surrounding spaces. Ordinary leading or trailing spaces introduced during import may be removed in a prepared column. Preserve the raw request in column A. In B2, =TRIM(A2) creates a normalized candidate. In C2, =LEN(A2) records the original length; in D2, =LEN(B2) records the prepared length. DOC-01 has six characters, while the supplied trailing-space version has seven.

TRIM also reduces repeated ordinary interior spaces. That is why its use must be justified by a format rule that does not permit such spaces, rather than applied indiscriminately to arbitrary names or codes. It also does not remove every character that appears visually blank. In particular, a nonbreaking space is not the same as an ordinary space. If the normalized key still does not match, inspect the source rather than repeatedly applying increasingly broad replacement functions.

Do not remove hyphens, convert a two-digit suffix to a number, or force an absent code into the nearest existing identifier. DOC-01 and DOC-1 are not equal under the practice rule. The fact that both suggest the number one does not establish that they identify the same record. An exact lookup should continue to fail for an unsupported spelling unless the source authority approves a correction. A no-match warning can be the correct final result.

After an allowed normalization, run the same duplicate and return-field checks used for an untouched key. Cleanup does not guarantee a unique valid record. Log both the transformation and its justification: original length 7, prepared length 6, ordinary trailing space removed under the registry format rule, resulting match DOC-01.

```
Prepared key: =TRIM(A2)
Raw length: =LEN(A2)
Prepared length: =LEN(B2)
Lookup after approved cleanup:
=XLOOKUP(B2,Registry!$A$2:$A$5,Registry!$B$2:$B$5,"UNREGISTERED",0)
```

E11-X1 Normalize keys only under an explicit identity rule — application

For raw keys " DOC-02 ", "DOC-1", and "DOC-99", apply only the allowed ordinary outer-space cleanup. Give each prepared key, whether the registry contains it, and the justified location or exception. Registry remains DOC-01 through DOC-04.

Submit: Three normalization records with both the raw and prepared identifier preserved.

12 Validate the matched return field, not just the key

Finding a key answers only one question: whether a record exists. It does not establish that every field in that record is complete. Registry's DOC-03 exists in A4, but B4 is genuinely empty because its shelf assignment has not been recorded. Reporting UNREGISTERED would misdiagnose the problem; returning an unexplained zero or empty-looking result would obscure the missing field. The useful message is Missing location for a registered document.

A lookup's not-found argument is designed for the no-match case. It is not a complete data-quality policy for the return column. Before retrieving a location, confirm that there is exactly one matching key and that the corresponding location field is populated. For this small controlled registry, COUNTIF counts matching keys and COUNTIFS can count matches whose location is empty. The supplied keys contain no wildcard characters, and the location cells contain ordinary shelf labels or true blanks.

The formula below implements the sequence: missing request, no registration, duplicate registration, missing return field, then retrieval. Each branch corresponds to an observable state. A practical alternative is to place the match count and location status in helper columns, which makes intermediate evidence easier to audit. Do not add a catch-all IFERROR around this entire sequence; an accidental broken range should still demand repair.

IFNA requires the same care. =IFNA(lookup,"UNREGISTERED") replaces any #N/A returned by its first argument, not just an error caused by an absent key. If an existing key's return cell deliberately contains =NA(), the lookup can propagate that #N/A and IFNA can falsely call the key unregistered. The handler knows the error value, not its business cause. Checking existence separately avoids confusing absent identity with an unavailable attribute.

In a controlled test copy, put =NA() in DOC-02's location cell and look up DOC-02. The key is still registered, but its field now carries an error that needs investigation. Restore Shelf-B afterward. This test is distinct from the genuine blank-field check and from a missing-key test.

```
=IF(A2="","Missing key",
 IF(COUNTIF(Registry!$A$2:$A$5,A2)=0,"UNREGISTERED",
 IF(COUNTIF(Registry!$A$2:$A$5,A2)>1,"Duplicate key review",
 IF(COUNTIFS(Registry!$A$2:$A$5,A2,Registry!$B$2:$B$5,"")=1,
 "Missing location",
 XLOOKUP(A2,Registry!$A$2:$A$5,Registry!$B$2:$B$5,"UNREGISTERED",0)))))
```

E12-X1 Validate the matched return field, not just the key — application

For DOC-03 and DOC-99, record the match count and correct exception. Separately, explain why wrapping a lookup of a registered key whose return cell is =NA() in IFNA(...,"UNREGISTERED") can be misleading. Do not permanently alter the registry.

Submit: Two exact classifications and a return-error diagnosis.

13 Choose the narrowest useful error-handling scope

An error handler is a boundary around an expression. The larger that expression, the more different causes can be collapsed into one substitute. For a known missing denominator, an input test describes the cause directly. For a missing exact-match key, a lookup's not-found argument expresses the expected exception. For a field conversion with a declared format, a visible parse warning can be appropriate around that conversion alone.

Compare three small formulas. =IFERROR(6/0,"Review") returns Review. =IFNA(6/0,"UNREGISTERED") leaves #DIV/0! visible because it handles #N/A, not division by zero. =IFNA(NA(),"Unavailable") returns Unavailable. The third formula does not repair the unavailable value; it changes how a known #N/A is presented. Whether that is appropriate depends on whether the audience still receives the underlying reason and whether downstream calculations can distinguish unavailable from numeric zero.

IFERROR handles several common error types, including #REF!, #VALUE!, #DIV/0!, #N/A, and #NAME?. That breadth can be useful at a carefully documented presentation boundary, but it is dangerous during diagnosis. If the expression contains both a lookup and a multiplication, a returned warning such as UNREGISTERED could conceal a broken reference or invalid text multiplier. A cause-specific message should be generated by a cause-specific check, not attached to a broad catch-all.

Use fault injection to test the boundary. In a disposable sheet, first create the expected exception. Then introduce a different fault, such as changing a valid reference to a broken one. If both states produce exactly the same apparently benign output, the handler may be masking evidence the user needs. Restore the test fault, and retain both observations.

A visible general label such as Calculation error is less misleading than zero, but still insufficient when it replaces the only diagnostic evidence. Retain the underlying formula result in an inspection area or leave it visible until repaired, and record the actionable cause. Do not claim that an error was resolved merely because a handler returned text successfully. Resolution means either a verified repair has restored the intended calculation or an accurately described expected exception remains openly unavailable.

Test expression	Expected displayed result	Meaning
=IFERROR(6/0,"Review")	Review	Broad replacement
=IFNA(6/0,"UNREGISTERED")	#DIV/0!	Unexpected error remains visible
=IFNA(NA(),"Unavailable")	Unavailable	Known #N/A presentation
=IFERROR(6/#REF!,0)	0	Broken structure concealed

E13-X1 Choose the narrowest useful error-handling scope — application

A colleague proposes =IFERROR(B2/C2,0) for every ratio. Test (B2,C2)=(0,5), (8,0), and (8,blank), then consider a broken denominator reference. Identify which outputs become indistinguishable and propose a narrower design.

Submit: A four-state comparison and a specific alternative that retains genuine zero, missing denominator, zero denominator, and structural error as distinct states.

14 Measure completeness before interpreting summaries

A row-level warning can disappear at the summary level even when no explicit IFERROR is used. SUM and AVERAGE over cell references can ignore text, so a column containing numbers and warning labels may produce a plausible numeric summary of only the numeric subset. That can be useful if the subset is clearly identified, but misleading if the result is described as covering all records.

On SummaryLab, place 10, 20, the text Missing input, and 0 in B2:B5. There are four records but only three numeric observations. SUM(B2:B5) is 30 and AVERAGE(B2:B5) is 10. The average is calculated over 10, 20, and 0, not over all four records. The measured zero must remain included. Replacing the warning with numeric zero changes the average to 7.5 and falsely treats the missing observation as measured.

Coverage can be stated as three valid observations out of four expected, or 75 percent. A suitable label is Mean of available observations: 10; coverage 3/4. A complete-population mean remains unknown without the fourth value. The number 10 is not intrinsically wrong; its interpretation is wrong if it is presented as the completed result for all four records.

For a strict release rule, withhold the headline until all required records are valid. In this small table, =IF(COUNT(B2:B5)=ROWS(B2:B5),AVERAGE(B2:B5),"Incomplete: review rows") performs a numeric-coverage gate. It assumes every row in the range is an expected observation and that numeric type alone defines validity. If the business rule also checks bounds or key status, gate on that validated status instead. A count is not a substitute for semantic validation.

Ratios need an additional interpretation check. Averaging per-row ratios gives equal weight to rows, while dividing the summed numerators by the summed denominators weights rows by denominator. For records 1/2 and 9/10, the mean of ratios is 0.7, but the combined ratio is 10/12, approximately 0.833333. Neither is a universal replacement for the other. Specify the intended summary and use matching eligible rows for both totals. Missing-data handling must not quietly change the population or the meaning of the metric.

```
Available sum: =SUM(B2:B5)
Available mean: =AVERAGE(B2:B5)
Numeric coverage: =COUNT(B2:B5)/ROWS(B2:B5)
Strict headline:
=IF(COUNT(B2:B5)=ROWS(B2:B5),AVERAGE(B2:B5),"Incomplete: review rows")
```

E14-X1 Measure completeness before interpreting summaries — application

For B2:B5 containing 6, 12, "Missing input", and 0, calculate the available sum, available mean, coverage, zero-masked mean, and strict-headline output. State which result is not yet known.

Submit: Five exact outputs and one clear statement about the unavailable complete-population mean.

15 Separate human-readable status from the numeric channel

A single column containing both numbers and warning text is convenient for reading but difficult to reuse safely. A clearer design provides one column for a human-readable status and another for a numeric result or an explicit unavailable error. The status explains why a row cannot be calculated; the numeric channel does not pretend that an absent measurement is zero.

On StatusLab, column B holds numerator and C holds denominator. Column D applies the arithmetic-only validation contract from Unit 5, but returns OK rather than the ratio when inputs pass. Column E uses =IF(D2="OK",B2/C2,NA()). A valid row has status OK and a number. An invalid row retains its specific status and shows #N/A in the numeric channel. Here #N/A is intentional evidence that a numeric value is unavailable, not proof that a lookup failed.

The deliberate NA() matters downstream. A simple SUM or AVERAGE over numeric results containing #N/A will not quietly ignore those rows as text warnings. The headline therefore requires an explicit release rule. For a complete-only average, =IF(COUNTIF(D2:D5,"OK")=ROWS(D2:D5),AVERAGE(E2:E5),"Incomplete: review status") checks all four expected statuses before presenting the metric. If an unexpected structural error affects the calculation despite an OK status, the error remains visible rather than being converted to a valid-looking number.

Do not create a circular dependency by making the status depend on the final headline or by making inputs read back from a diagnostic summary. Status should inspect the input contract directly; the numeric formula should use the validated inputs; the report should consume those results. An independent audit area may compare them, but it should not become the source of the business measurement.

For partial reporting, prepare an explicitly labeled subset using rows whose status is OK, and report its coverage. Do not overwrite unavailable numeric cells with zeros just to make a chart or total accept them. This course uses visible #N/A deliberately, so a reviewer should ask whether the accompanying status is accurate and whether the valid path still calculates, rather than demanding that every error symbol disappear.

```
D2 =IF(B2="","Missing numerator",
 IF(NOT(ISNUMBER(B2)),"Invalid numerator",
 IF(C2="","Missing denominator",
 IF(NOT(ISNUMBER(C2)),"Invalid denominator",
 IF(C2=0,"Zero denominator","OK")))))
```

```
E2 =IF(D2="OK",B2/C2,NA())
```

E15-X1 Separate human-readable status from the numeric channel — application

Create four rows with (numerator,denominator) equal to (6,3), (0,4), (8,0), and (blank,2). List every status and numeric-channel result. Apply the complete-only headline formula to the four rows.

Submit: Four status/result pairs and the exact headline text.

16 Design regression tests that challenge the repair

A regression test asks whether a repair still behaves correctly when relevant conditions change. A useful test set covers the stated categories, the boundaries between them, and at least one unexpected structural fault that the handler should not conceal. Tests need expected outcomes written before their results are observed.

The regression_cases.csv asset provides eight arithmetic-only cases for the Unit 5 guard. T01 uses numerator 8 and denominator 2; T02 uses 0 and 2; T03 uses 8 and a blank; T04 uses 8 and 0; T05 uses a blank numerator and 2; T06 uses the text eight and 2; T07 uses 8 and the text "2"; T08 uses 8 and 0.5. Their intended outcomes are 4, 0, Missing denominator, Zero denominator, Missing numerator, Invalid numerator, Invalid denominator, and 16 respectively.

T08 tests a small positive denominator rather than another ordinary integer. It confirms that the arithmetic-only contract accepts nonzero numeric values even when the resulting ratio exceeds one. This is not the final count-based business rule. Keeping the contract name with the test prevents a later reviewer from incorrectly reusing the expected value 16 in a workflow that prohibits completed counts above available slots. Tests are meaningful only relative to a clearly stated specification.

Add a fault test in a disposable formula cell by replacing a required reference with #REF!. The expected observation is a visible structural error, not a missing-input label or zero. That test examines the scope of the handler rather than the data rule. Never break the preserved source workbook merely to demonstrate the defect. Keep the faulty formula confined to the practice copy and restore the working formula afterward.

Record expected output, actual output, and pass or fail for each case. A text label should match the intended category, and a numeric output should be checked independently. For repeating decimals, state a tolerance or number of displayed decimal places instead of comparing rounded text to an exact fraction. The current regression cases use exact terminating results, so no tolerance is needed. When a test fails, preserve the failing input and formula as evidence; do not rewrite the expected answer to match the implementation unless the specification itself has been corrected.

E16-X1 Design regression tests that challenge the repair — application

Run T01–T08 from regression_cases.csv using the arithmetic-only guard. Add T09 with both inputs blank and T10 with numerator 8 and denominator -2. State the expected result under this unit’s contract, and identify which additional policy would reject T10.

Submit: A ten-row expected/actual test record and a note separating arithmetic validity from count-based business validity.

17 Keep a repair ledger that separates evidence from guesses

A useful repair ledger answers five questions: what failed, why it failed, what evidence authorized the change, what was changed, and how the result was verified. It should also say whether the underlying case is resolved or still waiting for information. A visible warning may be the correct finished handling of a case even when the missing data itself remains unresolved. Distinguish a completed diagnostic action from a completed source record.

The supplied source_tickets.csv contains four explicit corrections. T101 confirms ImportLab I03 is numeric 12. T102 confirms RatioLab R05 completed is numeric 12. T103 confirms RatioLab R06 slots is numeric 3. T104 confirms Capstone C06 completed is numeric 8. No ticket supplies RatioLab R03's missing denominator or R07's missing numerator. A nearby value, a pattern in other rows, or a desired total is not an alternative authority for those fields.

For R05, record the raw numerator twelve, the Invalid numerator status under the arithmetic-only guard, ticket T102, the prepared numeric 12, and the check 12/3=4. For R06, record the denominator's original text type, ticket T103, the prepared numeric 3, and 15/3=5. Preserve the raw file and make these changes only in the prepared working copy. Later business-range checks may still reject the resulting ratios; source correction and final eligibility are different stages.

For R03, record missing denominator, no supplied source correction, warning retained, and recovery test 18/6=3 performed only in a copy with the temporary input restored. The test proves that the formula can recover; it does not verify that 6 is the true denominator. This distinction is often lost when a test value makes a worksheet look complete. The ledger should explicitly identify temporary values so they cannot be mistaken for approved facts.

An effective entry names the record, field or formula, relevant ticket or specification, exact replacement, and independent expected result. If the action remains blocked, state the missing evidence rather than guessing a deadline or an owner not provided by the case.

E17-X1 Keep a repair ledger that separates evidence from guesses — application

Write ledger entries for R06 and R07. For R06 use T103. For R07 no correction ticket exists. Include a valid-path test for R07 using temporary numerator 2 and denominator 4, then restore its blank numerator.

Submit: Two evidence-linked ledger entries with exact results and honest unresolved status.

18 Debug a multistage calculation without moving the error

A calculation often combines input preparation, lookup, multiplication, and a final ratio. Wrapping the entire chain in one handler obscures which stage failed. Break the expression into meaningful steps that each own one responsibility.

On Pipeline, place document code DOC-02 in A2, quantity 8 in B2, and available slots 4 in C2. A small factor registry in H2:I3 contains DOC-01 with numeric factor 1.5 and DOC-02 with numeric factor 2. In D2, retrieve the exact factor. In E2, calculate quantity times factor. In F2, divide that weighted quantity by slots. The valid chain is factor 2, weighted quantity 16, and final ratio 4. These values are synthetic workload units, not prices or financial forecasts.

Now change only A2 to DOC-99 in a copy. The first failure is the absent key at D2. If D2 returns text UNREGISTERED and E2 still multiplies B2*D2, E2 produces a text-arithmetic error. That downstream #VALUE! is secondary; correcting B2 would not fix the missing registry relationship. A properly designed status flow would keep the no-match reason visible and make dependent numerical outputs unavailable rather than trying to multiply the label.

Consider a different test: restore DOC-02, but change its factor cell I3 to the text two. The key exists, so UNREGISTERED is the wrong diagnosis. The return-field type is invalid. Restore numeric 2 before testing C2=0, which creates a third independent failure at the final denominator stage. Diagnose the earliest stage whose own contract fails.

A final report can use a combined readiness status, but the component evidence should remain inspectable. Check key existence and uniqueness, factor type, quantity type, and denominator validity before computing the final ratio. If multiple inputs are wrong, state whether the status shows the first failure or all failures. When repairing one stage, rerun the other test cases to make sure the change did not merely move the error into a downstream cell. End with the valid chain restored and verify the three numerical outputs independently.

```
Pipeline!D2 =XLOOKUP(A2,$H$2:$H$3,$I$2:$I$3,"UNREGISTERED",0)
Pipeline!E2 =B2*D2
Pipeline!F2 =E2/C2
These unguarded formulas are diagnostic specimens, not the release design.
```

E18-X1 Debug a multistage calculation without moving the error — application

For three separate Pipeline copies, test an absent DOC-99 key, a registered DOC-02 whose factor is text two, and a zero final denominator with all other inputs valid. Name the earliest failing stage and justified reason. Then use quantity 6, factor 2, and slots 3 to verify recovery.

Submit: Three stage diagnoses and the recovered factor, weighted quantity, and ratio.

19 Resolve a mixed queue while preserving independent exceptions

The Capstone dataset combines the course's main error-handling problems without using private records. Columns A through D are Record, Completed, Slots, and Document. Rows 2 through 9 contain C01 through C08. The registry is unchanged from Unit 10, including DOC-03's missing location. Ticket T104 is the only authorized input correction: C06's word eight may be prepared as numeric 8. Do not invent values for any other unresolved field.

This capstone adds the full count-based business contract. Completed must be present, numeric, and nonnegative. Slots must be present, numeric, and positive. Completed cannot exceed Slots. Evaluate those checks in that order, using the status labels shown below. Document status is checked independently for missing key, no match, duplicate key, and missing location. A row is ready only when its ratio status is OK and its document lookup produces a valid shelf. This independence matters because one row can contain more than one defect.

Create E as Ratio status, F as Ratio value, G as Document status or location, and H as Ready status. For F use =IF(E2="OK",B2/C2,NA()). For this supplied registry, H can use =IF(AND(E2="OK",LEFT(G2,6)="Shelf-"),"READY","REVIEW"). The Shelf- test is a case-specific shortcut because every permitted location in this registry has that prefix; it is not a general validation method for arbitrary workplace data. The document checks remain those taught earlier.

Do not force a single warning to stand for every defect. C05 has completed 7 and slots 4, as well as the unregistered key DOC-99. Its ratio status and document status should both remain visible. C07 has an arithmetically and operationally valid ratio, but its registered document lacks a location. A report that marks every numerically valid row ready would miss that operational exception. A report that blanks the ratio whenever the location is missing would erase useful valid evidence.

Compute coverage separately for valid ratios and fully ready records. If a partial combined ratio is requested for READY records, sum completed and slots over the same eligible rows, state the population explicitly, and retain the overall incomplete status. Do not average individual row ratios unless that is the requested metric.

```
Ratio status labels in precedence order:
Missing numerator; Invalid numerator; Negative numerator;
Missing denominator; Invalid denominator; Zero denominator;
Negative denominator; Completed exceeds slots; OK.
Full executable status formula is in assets/cell_map.txt.
```

Use all eight rows in assets/capstone.csv. Worked diagnostic excerpt: C05 has Completed=7, Slots=4, and Document=DOC-99; C07 has Completed=6, Slots=12, and Document=DOC-03. Keep the full source unchanged.

E19-X1 Resolve a mixed queue while preserving independent exceptions — application

Prepare C06 using T104, then produce all eight ratio statuses, numeric results, document outcomes, and READY/REVIEW labels. Count valid ratios and READY records. Calculate the combined ratio only for READY records and label its coverage. State whether a complete eight-record headline may be released.

Submit: An eight-row exception queue, two coverage counts, a matched-population partial ratio, and a release decision.

20 Review the repair as an evidence package

The final review should establish that the model is understandable, reproducible, and honest about what remains unknown. Review the input contract, exact formulas, verified source corrections, warning meanings, dependency tests, and coverage statement together. Each piece answers a different question: whether the data is authoritative, whether the calculation follows the rule, whether exceptions remain visible, and whether the reported population matches the available evidence.

Use the Capstone output as the release specimen. A good handover identifies the only authorized source correction, C06 from text eight to numeric 8 under T104. It states that C03, C04, C05, C07, and C08 remain under review, while C01, C02, and C06 are ready. It preserves C07's valid ratio separately from its missing location, and preserves both of C05's defects. These details show that the learner has not collapsed all problems into one cosmetic label.

Inspect at least one structural recovery test and one genuine-zero test. A broken reference must remain detectable; a legitimate numerator of zero with a valid positive denominator must still produce numeric zero. Check an unregistered key and an existing key with an unavailable return field, because their surface symptoms can be confused by IFNA. Confirm that temporary test values have been restored and that the original raw assets remain unchanged.

The handover must distinguish completed repairs from unresolved source work. It is acceptable to say that a formula now handles missing data correctly while the data itself remains unavailable. It is not acceptable to say all errors fixed if the only change was substituting zeros. Equally, intentionally displaying #N/A in a numeric channel is not a failed assignment when the adjacent status accurately identifies the unavailable condition.

Finally, describe exactly which headline can be supported. The READY subset has three records, completed total 12, slots total 23, and combined ratio 12/23. It is not the completed result for all eight records. If the report uses a rounded percentage, retain the underlying fraction or full precision in the calculation. The recipient should be able to distinguish rounding from uncertainty.

E20-X1 Review the repair as an evidence package — application

Review this proposed handover: "All eight records are fixed. Missing results were set to zero. The final ratio is 52.17%." Replace it with a factual handover of 90–140 words using the Capstone evidence. Include the correction, unresolved records, two coverage counts, the subset ratio, and the limits of release.

Submit: One concise evidence-based handover. Do not invent resolutions, owners, or future completion dates.

Assessment

Choose one best answer per question. Questions 1–10 check the foundation concepts; questions 11–20 apply the extended study units. Complete and submit the practical tasks separately. A strong quiz score does not replace correct practical work.

1. A formula contains =B2*#REF! after an input column was removed. Which error identifies the invalid reference?

A. #REF!

B. #DIV/0!

C. #NAME?

D. #N/A

2. In original case E2, B2=12 and C2 is genuinely blank. What should the ratio show before a verified denominator is supplied?

A. 0, treating the blank as measured zero.

B. 4, using the test denominator 3 permanently.

C. Zero denominator, treating every blank as numeric zero.

D. Missing denominator.

3. In original case E3, B2=12 and C2 is numeric 0. Which result preserves the meaning of this input condition?

A. 0, because zero is present in the denominator.

B. Missing denominator, because no ratio can be calculated.

C. Zero denominator.

D. 12, ignoring the invalid divisor.

4. Why is blanket IFERROR(...,0) risky when applied to a mixed error queue?

A. It restores deleted source references without recording them.

B. It can make different failures look like genuine numeric zeros.

C. It replaces only lookup no-match errors.

D. It proves every remaining numeric output follows the specification.

5. Original E1 specifies quantity B2=8 multiplied by shared rate H2=0.5. D2 contains =B2*#REF!. Which repair is justified?

A. Replace D2 with =IFERROR(B2*#REF!,0).

B. Reconnect D2 to =B2*$H$2 and verify 4.

C. Use the adjacent H3 value because its reference is valid.

D. Enter 4 as a fixed output and remove the formula.

6. Original E4 requests DOC-99; the verified registry contains only DOC-01 and DOC-02. What is the justified lookup result?

A. Shelf-B, because DOC-02 is the nearest registered code.

B. The location copied from the preceding request.

C. UNREGISTERED.

D. A new DOC-99 registration with a guessed shelf.

7. Why may original E5’s text quantity ten be prepared as numeric 10?

A. Any word in a quantity column may be converted by guesswork.

B. The supplied verified source explicitly confirms numeric 10.

C. The multiplication error proves the intended quantity was 10.

D. The required output 30 can be used to reverse-engineer any missing input.

8. Original E6 uses =SUMM(B2:B4) for the values 2, 3, and 4. What is the cause of #NAME? in this expression?

A. SUMM is misspelled; SUM is the intended function.

B. The range contains a zero denominator.

C. One of the three numbers must be an unregistered key.

D. A valid result of 9 cannot be represented numerically.

9. Which test verifies original E2/E3’s valid ratio path without inventing permanent source data?

A. Check only that the warning text is formatted consistently.

B. Replace each formula with a manually typed 4.

C. Temporarily use denominator 3, verify 12/3=4, then restore the original input.

D. Leave denominator 3 in both rows because the formula now works.

10. Does a workbook with no visible error code necessarily calculate the intended quantities?

A. No; a valid reference to the wrong input can still produce a plausible number.

B. Yes; all numeric outputs satisfy the business specification.

C. Yes; applying IFERROR proves every underlying dependency is correct.

D. Yes; a consistent formula pattern proves the intended operation is correct.

11. B2 contains the formula =""". Which description is correct under the course missing-data policy?

A. ISBLANK(B2) is TRUE, and B2 contains a measured zero.

B. ISBLANK(B2) is FALSE, but B2="" is TRUE and the value is treated as missing.

C. ISNUMBER(B2) is TRUE because the formula calculated successfully.

D. B2="" is FALSE because every formula result is numeric.

12. B2:B4 contain 10, 20, 30. Only H2=0.4 is the authorized shared rate; H3=0.1 and H4=0.9 are unrelated notes. Copying =B2*H2 down gives 4, 2, 27. Which repair follows the specification?

A. Keep the formulas because every result is numeric.

B. Wrap every result in IFERROR(...,0).

C. Use =B2*$H$2 and fill down, producing 4, 8, 12.

D. Change H3 and H4 to 0.4 without changing the relative references.

13. ImportLab identifies the text token "1.250,50" as comma-decimal and dot-grouping format. Which preparation preserves its stated numeric meaning?

A. =NUMBERVALUE(B3,",",".") giving 1250.5.

B. =NUMBERVALUE(B3,".",",") regardless of the declared format.

C. Remove both punctuation characters and use 125050.

D. Replace the token with 0 because it was imported as text.

14. DuplicateLab contains DOC-02 twice, once with Shelf-B and once with Shelf-Z. A forward XLOOKUP returns Shelf-B. What should the release status be?

A. READY because XLOOKUP returned a text location.

B. UNREGISTERED because the two shelves disagree.

C. READY after replacing the second shelf with the first.

D. Duplicate key review because the match count is 2.

15. Registry contains DOC-03 exactly once, but its location field is empty. Which output accurately identifies the problem?

A. UNREGISTERED.

B. Shelf-A copied from the previous record.

C. Missing location.

D. 0, because the key was found.

16. What does =IFNA(6/0,"UNREGISTERED") display?

A. UNREGISTERED.

B. #DIV/0!.

C. 0.

D. #N/A.

17. B2:B5 contain 6, 12, the text Missing input, and numeric 0. Which summary accurately describes AVERAGE(B2:B5)?

A. 4.5, the complete-population mean of four measured records.

B. 9, because all zero values should be excluded.

C. 6, the complete-population mean with no limitation.

D. 6, the mean of three numeric observations, with 75% coverage.

18. A two-channel design uses status D2 and numeric result =IF(D2="OK",B2/C2,NA()). If D2 is Missing denominator, which interpretation is correct?

A. #N/A intentionally marks the numeric result unavailable; D2 explains why.

B. #N/A proves that a document code was not found.

C. The numeric result should be changed to 0 to make it usable.

D. D2 must be changed to OK because NA() is itself an error.

19. After the authorized C06 correction, Capstone has valid ratios for C01, C02, C06, and C07. Only C01, C02, and C06 also have valid document locations. What are the two coverage counts?

A. 4/8 valid ratios and 4/8 READY.

B. 4/8 valid ratios and 3/8 READY.

C. 3/8 valid ratios and 3/8 READY.

D. 8/8 valid ratios and 3/8 READY because warnings count as ratios.

20. The READY records have (completed, slots) of (4,8), (0,5), and (8,10). Which result is their combined completed-to-slots ratio?

A. (0.5+0+0.8)/3, approximately 43.33%.

B. 12/8=150%, because eight source records were supplied.

C. 12/23, approximately 52.17%, labeled as the three-record READY subset.

D. 12/23, approximately 52.17%, labeled as the complete eight-record result.

Foundation completion criteria

Practical pass: all six causes identified and appropriately repaired or labeled; missing and zero denominator remain distinct; unregistered codes stay visible; no blanket blank or zero masking. Quiz target: 8 of 10.

Sources and further reading

Microsoft Excel: IFERROR function
https://support.microsoft.com/en-us/excel/functions/iferror-function

Microsoft: Detect formula errors in Excel
https://support.microsoft.com/en-gb/excel/detect-formula-errors-in-excel

Microsoft Excel: XLOOKUP function
https://support.microsoft.com/en-us/excel/functions/xlookup-function

IFNA function
https://support.microsoft.com/en-us/excel/functions/ifna-function

IS functions
https://support.microsoft.com/en-us/excel/functions/is-functions

NUMBERVALUE function
https://support.microsoft.com/en-us/excel/functions/numbervalue-function

TRIM function
https://support.microsoft.com/en-us/excel/functions/trim-function

AVERAGE function
https://support.microsoft.com/en-us/excel/functions/average-function

SUM function
https://support.microsoft.com/en-us/excel/functions/sum-function

NA function
https://support.microsoft.com/en-us/excel/functions/na-function

Switch between relative, absolute, and mixed references
https://support.microsoft.com/en-us/excel/switch-between-relative-absolute-and-mixed-references

How to avoid broken formulas in Excel
https://support.microsoft.com/en-us/excel/how-to-avoid-broken-formulas-in-excel

Fix an inconsistent formula
https://support.microsoft.com/en-us/excel/fix-an-inconsistent-formula

Count how often a value occurs in Excel
https://support.microsoft.com/en-us/excel/count-how-often-a-value-occurs-in-excel

COUNTIFS function
https://support.microsoft.com/en-us/excel/functions/countifs-function

SUMIF function
https://support.microsoft.com/en-us/excel/functions/sumif-function

LEN function
https://support.microsoft.com/en-us/excel/functions/len-function

COUNT function
https://support.microsoft.com/en-us/excel/functions/count-function?nochrome=true

Look up values with VLOOKUP, INDEX, or MATCH
https://support.microsoft.com/en-us/excel/look-up-values-with-vlookup-index-or-match