

# Lookup Functions Practice Workbook Guide

Four Exercises · Healthcare IT Scenarios ·  
TechFitFlow.com

Each exercise shows the same lookup task solved with VLOOKUP, XLOOKUP, and INDEX/MATCH. The goal is to see where each function excels and where each fails in the same scenario.

## EXERCISE 1 — ICD-10 CODE MAPPING (BASIC LOOKUP)

**Scenario:** Encounter export from Epic Clarity has DiagCode column. CodeMap Table has Code (col A) and Description (col B). Add Description to each encounter row.

### VLOOKUP

```
=IFERROR(VLOOKUP([@DiagCode],CodeMap,2,FALSE),"Code not in reference")
```

Works correctly. BUT: insert a column into CodeMap between Code and Description and col\_index 2 now returns the wrong column — silently.

### XLOOKUP

```
=XLOOKUP([@DiagCode],CodeMap[Code],CodeMap[Description],"Code not in reference")
```

Best option on M365. Direction-independent, built-in fallback, immune to column insertion. If colleague opens in Excel 2019 — shows #NAME?.

### INDEX/MATCH

```
=IFNA(INDEX(CodeMap[Description],MATCH([@DiagCode],CodeMap[Code],0)),"Code not in reference")
```

Works in all Excel versions. Immune to column insertion. Use IFNA not IFERROR so structural errors remain visible.

## EXERCISE 2 — TWO-CONDITION PAYER-SERVICE RATE LOOKUP

**Scenario:** RateTable has Payer, ServiceType, Rate columns. Each payer-service combination has a unique rate. Look up rate where Payer=BCBS AND ServiceType=Inpatient for each claim row.

### VLOOKUP

NOT POSSIBLE without a helper column in RateTable.

VLOOKUP cannot match on two conditions. A helper column =[@Payer]&"|"&[@ServiceType] in RateTable would allow it — but requires modifying the rate table.

### XLOOKUP

Option A (helper column): =XLOOKUP([@Payer]&"|"&[@ServiceType],RateTable[Key],RateTable[Rate],"Not found")  
Option B (nested): =XLOOKUP([@Payer],RateTable[Payer],XLOOKUP([@ServiceType],RateHeaders,RateMatrix),"Not found")

Requires either a helper column in RateTable or a nested XLOOKUP against a matrix layout. Both work; the helper column approach is more readable.

### INDEX/MATCH

```
=IFERROR(INDEX(RateTable[Rate],MATCH(1,(RateTable[Payer]=[@Payer])*(RateTable[ServiceType]=[@ServiceType]),0),"Rate not found")
```

No helper column needed. Boolean array multiplication finds the row where both conditions are TRUE. Ctrl+Shift+Enter in pre-M365 Excel.

### EXERCISE 3 — MOST RECENT MATCH: UAT RETEST STATUS

**Scenario:** TestLog Table logs each test run including retests. ScenarioID SC-042 appears 3 times: initial Fail, retest Fail, retest Pass. Summary needs the most recent status.

#### VLOOKUP

```
=VLOOKUP(A2,TestLog,3,FALSE)
```

Returns first match only — the initial test result (Fail), not the most recent retest result (Pass). Wrong output for retest tracking workbooks.

#### XLOOKUP

```
=XLOOKUP(A2,TestLog[ScenarioID],TestLog[Status],"Not found",0,-1)
```

search\_mode=-1 searches last-to-first. Returns the most recent entry for SC-042 (Pass). Clearest solution for this use case.

#### INDEX/MATCH

```
=IFERROR(INDEX(TestLog[Status],MATCH(2,1/(TestLog[ScenarioID]=A2),1)),"Not found")
```

Workaround for last match. More complex to write and maintain. Use XLOOKUP if M365 is available for this scenario.

### EXERCISE 4 — IFNA VS IFERROR: ERROR HANDLING COMPARISON

**Scenario:** CodeMap VLOOKUP has a #REF! structural error on row 15 (CodeMap reference broken). Compare what IFERROR and IFNA return for row 15.

#### IFERROR wrapping — wrong approach

```
=IFERROR(VLOOKUP([@DiagCode],CodeMap,2,FALSE),"Code not in reference")
```

Row 15 returns "Code not in reference" — hides the structural #REF! error. The workbook looks complete but a formula is broken. The broken reference is invisible.

#### IFNA wrapping — correct approach

```
=IFNA(VLOOKUP([@DiagCode],CodeMap,2,FALSE),"Code not in reference")
```

Row 15 returns #REF! — the structural error is visible. You can identify and fix the broken CodeMap reference. Other rows that return #N/A (code not found) still show "Code not in reference".

#### XLOOKUP built-in fallback — also correct

```
=XLOOKUP([@DiagCode],CodeMap[Code],CodeMap[Description],"Code not in reference")
```

if\_not\_found catches only #N/A. A structural error (#REF! in CodeMap) still surfaces as #REF! in the XLOOKUP result — same behaviour as IFNA.

