

# How to Group Times into Unequal Buckets in Excel

Authored by  
**mohammed loot**

January 4, 2026

## RECOMMENDED CITATION

mohammed loot (2026). *How to Group Times into Unequal Buckets in Excel*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=124600>

Grouping chronological data is a fundamental requirement in professional data analysis, especially when working with time-series information such as operational logs, shift schedules, or customer interaction times. While standard tools often offer fixed interval grouping (e.g., hourly or daily bins), real-world scenarios frequently demand the creation of **unequal buckets**. This approach is essential because activity levels are rarely uniform across a 24-hour cycle. For instance, peak business hours (10 AM to 1 PM) might require finer granularity than slow overnight periods (12 AM to 6 AM). Excel, being a versatile tool, allows sophisticated customization to handle these specific requirements, moving beyond simple pivot table groupings to apply logical conditions based on varying time segments.

The ability to define these custom, unequal time intervals--often referred to as 'buckets' or 'bins'--enables analysts to perform more accurate and meaningful segmentation of data analysis. When dealing with large datasets that contain a mixture of short and long time intervals, standard, fixed-width grouping can obscure critical patterns. By contrast, tailoring the bucket size to reflect genuine operational significance allows for focused scrutiny on high-traffic periods while efficiently aggregating less active times. This technique, relying primarily on logical functions and time conversion utilities within Excel, provides a powerful yet accessible method for advanced data manipulation.

## Understanding How Excel Handles Time Values

Before diving into the grouping formula, it is critical to understand how Excel stores and calculates time. Unlike human-readable formats (like "10:00 AM"), Excel treats time as a fraction of a 24-hour day. Midnight (12:00:00 AM) corresponds to 0, and noon (12:00:00 PM) corresponds to 0.5. Therefore, a time value like 6:00 AM is stored internally as 0.25 (one quarter of a day). When comparing times or applying logical tests, Excel uses these underlying decimal values. This fundamental concept is crucial because our grouping mechanism will depend on comparing the raw time values in our dataset against the fractional equivalents of our defined bucket boundaries.

To facilitate accurate comparison between textual time inputs (like "10:00:00 AM") and the numerical time values in a dataset, we must employ the TIMEVALUE function. This function converts a text string representing time into the corresponding fractional number that Excel uses for calculations. Without this conversion, Excel would struggle to accurately evaluate the logical conditions necessary for assigning data points to their respective unequal buckets. For example, the text "10:00:00 AM" becomes approximately 0.416667 when processed by the TIMEVALUE function, allowing for precise numerical comparison against the stored time data.

Furthermore, ensuring that the source data is formatted correctly is paramount. While the display format might show hours, minutes, and seconds, the underlying cell format must be recognized by Excel as a valid time or date-time serial number. If the data is imported as plain text, using

functions like TIMEVALUE function will become indispensable to coerce the text into a comparable numerical format. This preparedness ensures that the complex logical structure we are about to build operates on reliable, comparable numerical data, maximizing the efficiency and correctness of the unequal time grouping process.

## Constructing the Logic: Nested IF Functions

The primary mechanism for grouping times into custom, unequal buckets in Excel is the utilization of nested functions, specifically the IF function. The IF function allows us to establish a sequence of conditional tests: if the first condition is met, a specific bucket label is assigned; if it is not met, the function proceeds to the next condition, thus creating distinct, mutually exclusive time ranges. This is particularly effective for defining unequal intervals because each subsequent logical test automatically assumes the failure of the preceding tests, simplifying the logic required for range definition.

When defining these intervals, it is crucial to arrange the logic sequentially, typically starting from the lowest time boundary and progressing upwards, or vice versa. Since we are using an inclusive upper boundary (e.g., less than or equal to a specific time), the order ensures that a time value falls into the narrowest and earliest applicable bucket. For example, if we test for " $\leq$  10 AM" first, any time before 10 AM is captured immediately. If the time is later than 10 AM, the function proceeds to the next nested IF function, which might test for " $\leq$  1 PM." This systematic progression is the essence of creating non-overlapping, unequal time segments using nested functions.

The general syntax involves repeatedly embedding one IF function within the 'value\_if\_false' argument of the previous one. The final 'value\_if\_false' argument of the outermost IF statement serves as the catchment for all remaining times--typically the latest and largest time bucket. This structure provides flexibility; while the example below utilizes three buckets, you can use as many layers of nested IF statements as necessary to accommodate complex business requirements, remembering that Excel has limitations on the total depth of nesting allowed.

## The Core Formula Structure

You can use the following basic syntax in Excel to group times into unequal buckets, which combines the power of the TIMEVALUE function for numerical comparison and the nested IF structure for sequential evaluation:

The formula below assumes the time value being evaluated is located in cell **B2**:

```
=IF(B2<=TIMEVALUE("10:00:00 AM"),"12AM-10AM",IF(B2<=TIMEVALUE("1:00:00 PM"),"10AM-1PM","1PM-12AM"))
```

This particular formula looks at the time in cell **B2** and returns one of three categorical values based on sequential evaluation of the criteria:

**12AM-10AM** if the time value in cell **B2** is less than or equal to 10:00:00 AM. This is the first logical test, capturing all early morning times.

Else, **10AM-1PM** if the time value in cell **B2** is less than or equal to 1:00:00 PM. Importantly, because the first condition failed (meaning the time is already greater than 10 AM), this test effectively captures the interval from 10:00:01 AM up to 1:00:00 PM.

Else, **1PM-12AM**. This final catch-all condition assigns this bucket label to any time value that did not satisfy the two preceding conditions (i.e., times greater than 1:00:00 PM).

Note that this particular formula creates three distinct time buckets. The flexibility of nested functions allows you to use more layers of the **IF** function as needed to define finer or more numerous unequal time segments, depending on the requirements of your data analysis.

### Example: Grouping Employee Shift Times

To illustrate the practical application of this technique, consider a scenario involving human resources or operations management where we track the exact start times of employee shifts. We aim to categorize these start times into predefined, unequal buckets to better understand staffing patterns or compliance with scheduling policies. Suppose we have the following raw dataset in Excel, showing employee IDs and their respective shift commencement times in column B:

This dataset represents raw, granular data which is difficult to interpret in aggregate. By grouping these times, we can summarize the distribution of shift starts across the operational day.

	A	B	C	D	E	F
1	<b>Employee</b>	<b>Start Time</b>				
2	A001	12:33 AM				
3	A002	1:15 AM				
4	A003	2:30 AM				
5	A004	3:00 AM				
6	A005	8:14 AM				
7	A006	11:34 AM				
8	A007	12:55 PM				
9	A008	3:19 PM				
10	A009	6:17 PM				
11	A010	9:34 PM				
12						
13						
14						
15						
16						
17						
18						
19						

Our goal is to classify each start time based on specific strategic intervals. These intervals are chosen because they represent distinct periods of operational importance--for example, pre-business hours, core business hours, and late/post-business hours. We have strategically selected the following unequal time buckets for this categorization:

**12AM-10AM** (A long, early period capturing initial and early morning shifts.)

**10AM-1PM** (A shorter, focused period likely representing peak mid-day traffic or standard shift starts.)

**1PM-12AM** (A long period covering afternoon, evening, and late-night shifts.)

We can type the following formula into cell **C2** to initiate the categorization process:

```
=IF(B2<=TIMEVALUE("10:00:00 AM"),"12AM-10AM",IF(B2<=TIMEVALUE("1:00:00 PM"),"10AM-1PM","1PM-12AM"))
```

Once the formula is entered in C2, we must apply it to the entire dataset. Instead of manually retyping or copying, the most efficient method is to click on the small green square (the fill handle) at the bottom right corner of cell C2 and drag this formula down to the last row containing data.

This action automatically adjusts the cell reference (B2 changes to B3, B4, and so on) thanks to Excel's relative referencing feature, ensuring every shift time is evaluated against the fixed time boundaries defined in the formula.

	A	B	C	D	E	F
1	<b>Employee</b>	<b>Start Time</b>	<b>Time Bucket</b>			
2	A001	12:33 AM	12AM-10AM			
3	A002	1:15 AM	12AM-10AM			
4	A003	2:30 AM	12AM-10AM			
5	A004	3:00 AM	12AM-10AM			
6	A005	8:14 AM	12AM-10AM			
7	A006	11:34 AM	10AM-1PM			
8	A007	12:55 PM	10AM-1PM			
9	A008	3:19 PM	1PM-12AM			
10	A009	6:17 PM	1PM-12AM			
11	A010	9:34 PM	1PM-12AM			
12						
13						
14						
15						
16						
17						

## Interpreting the Categorized Results

As demonstrated by the resulting table, Column C now clearly shows the time bucket into which each employee's shift start time falls. This transformation simplifies subsequent data analysis, allowing summary statistics (like counts or averages) to be generated easily based on these custom categories, rather than on individual time stamps. The grouping structure is rigorously applied according to the sequential logic of the nested functions.

For example, observe the following specific results extracted from the categorized output, which highlight how the unequal boundaries operate based on the sequential logic of the IF function:

Employee A001 started working at **12:33 AM**. Since 12:33 AM (0.0231) is numerically less than the threshold for 10:00 AM (0.4167), they fell into the first time bucket: **12AM-10AM**.

Employee A006 started working at **11:34 AM**. This time is greater than 10:00 AM (failing the first IF test), but it is less than or equal to 1:00 PM (0.5417), placing them into the second time bucket: **10AM-1PM**.

Employee A008 started working at **3:19 PM** . This time failed both the first ( $\leq 10$  AM) and the second ( $\leq 1$  PM) IF tests, causing the formula to assign the final default category: **1PM-12AM**.

This resulting column of categorical data can now be used as the primary field for generating pivot tables, creating charts, or conducting detailed reports on resource allocation and operational efficiency, proving the utility of defining time intervals that reflect real-world operational rhythms rather than arbitrary fixed units.

## Handling Edge Cases and Data Integrity

When implementing time-based grouping, it is crucial to consider edge cases, particularly how the boundaries are defined. By using the less than or equal to operator ( $\leq$ ), we ensure the boundary time itself (e.g., 10:00:00 AM) is included in the current bucket, which is generally the desired behavior. However, analysts must be mindful of precision; if time data includes milliseconds, ensuring the TIMEVALUE function comparisons account for the required decimal precision is vital to prevent misclassification.

Data integrity also plays a significant role. If the source column (Column B in our example) contains text strings that Excel cannot interpret as valid time values, the TIMEVALUE function will return an error (usually #VALUE!), halting the grouping process. Best practices dictate performing a data validation check on the source column first, converting all entries to standardized Excel time format before applying the complex conditional logic. This preliminary step guarantees that the nested IF function executes smoothly across the entire dataset.

## Alternative Grouping Methods for Unequal Buckets

While nested IF statements are a powerful and direct method, they can become unwieldy and difficult to audit if the number of unequal buckets exceeds four or five. For situations requiring highly complex or frequently changing time boundaries, alternative Excel techniques offer better scalability and maintainability. One common alternative is using the **VLOOKUP** function (or its modern equivalent, XLOOKUP) with an approximate match type (Range Lookup set to TRUE).

To use VLOOKUP for unequal grouping, the analyst would create a separate reference table containing the lower bound of each time bucket and the corresponding bucket name. Because VLOOKUP with an approximate match searches for the largest value that is less than or equal to the lookup value, it inherently handles the sequential assignment of unequal ranges without requiring extensive nesting. This external reference table makes it vastly easier to modify the time boundaries without editing the core calculation formula, drastically improving the efficiency of the overall data analysis workflow, especially in environments where requirements are fluid.

## Further Resources for Excel Time Manipulation

Mastering the nested IF approach for unequal time bucketing is a significant step in advanced Excel functionality. For those seeking to further enhance their data handling capabilities, especially concerning time and date data, exploring related tutorials can be highly beneficial. Understanding other common tasks involving time allows for greater flexibility and sophistication in reporting and modeling efforts.

The following tutorials explain how to perform other common tasks in Excel:

ARABPSYCHOLOGY.COM