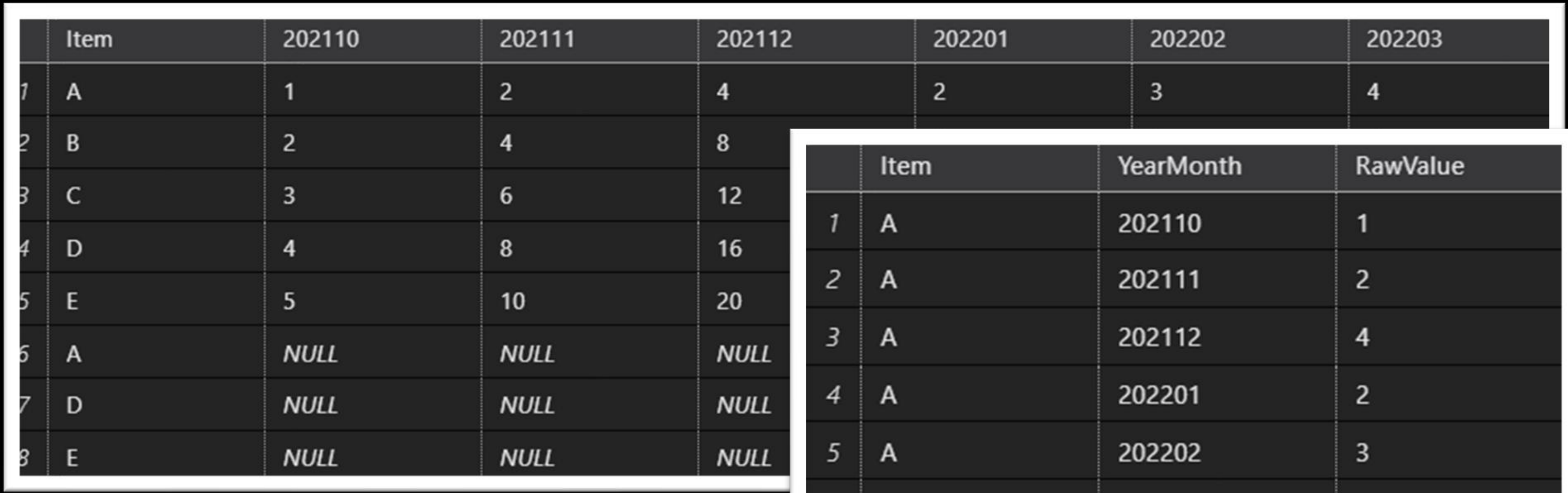
Certain tasks are easier in some languages

Choose wisely!

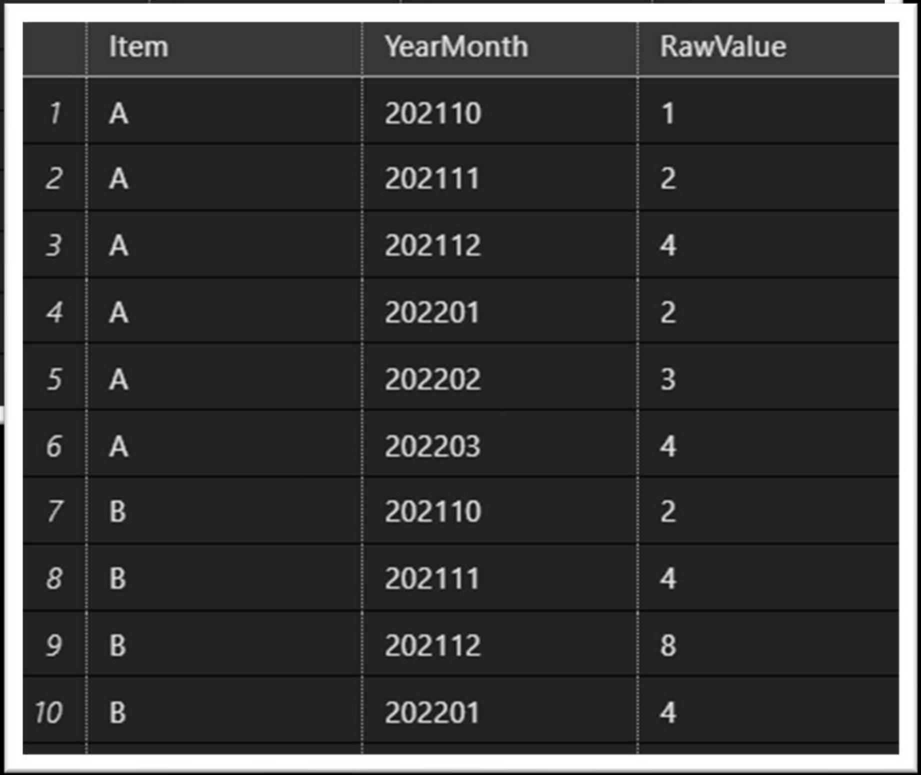
Unpivoting is a common operation

Change this



	Item	202110	202111	202112	202201	202202	202203
1	A	1	2	4	2	3	4
2	B	2	4	8			
3	C	3	6	12			
4	D	4	8	16			
5	E	5	10	20			
6	A	*NULL*	*NULL*	*NULL*			
7	D	*NULL*	*NULL*	*NULL*			
8	E	*NULL*	*NULL*	*NULL*			

To this



	Item	YearMonth	RawValue
1	A	202110	1
2	A	202111	2
3	A	202112	4
4	A	202201	2
5	A	202202	3
6	A	202203	4
7	B	202110	2
8	B	202111	4
9	B	202112	8
10	B	202201	4

Let’s compare:

SQL Server

PostgreSQL

Excel

M

Python

R

SQL (SQL Server)

Select the result of the UNPIVOT into a variable

Concatenate the non-ID column names with a dynamic UNPIVOT query

```sql
DECLARE @sql nvarchar(4000);

SELECT @sql =
'SELECT Item, YearMonth, RawValue
FROM source
UNPIVOT
(RawValue FOR YearMonth IN (' + cols + ')) unpvt'
FROM (
    SELECT STRING_AGG(QUOTENAME(COLUMN_NAME), ', ') as cols
    FROM INFORMATION_SCHEMA.COLUMNS
    WHERE TABLE_NAME = 'source'
    AND COLUMN_NAME <> 'Item'
) c;

EXEC(@sql);
```

Retrieve non-ID column names from INFORMATION_SCHEMA, then aggregate into a single comma-separated value

Execute the dynamic SQL

SQL (PostgreSQL)

```
WITH
json_rows AS
(SELECT
    row_to_json(df) AS row_json
FROM df),

item_tuples AS
(SELECT
    row_json->>'item' AS item,
    json_each_text(row_json) AS tuple
FROM json_rows)

SELECT
    item,
    (tuple).key AS yearmonth,
    (tuple).value
FROM item_tuples
WHERE (tuple).key != 'item';
```

Convert each row in the table to a JSON object with a {key:value} pair per column/row

Extract the ID column and produce one row per {key:value} pair in the JSON

Extract the key and value into separate columns

Excel

Data without the header row

```
=LET(
    dat, A1:G9,
    df, DROP(dat, 1),
    fn, LAMBDA(ym,
            HSTACK(
                TAKE(df, , 1),
                EXPAND(ym, ROWS(df), 1, ym),
                FILTER(df, TAKE(dat, 1) = ym)
            )),
    REDUCE(
        fn(INDEX(dat, 1, 2)),
        DROP(TAKE(dat, 1), , 2),,
        LAMBDA(a, b, VSTACK(a, fn(b)))
    )
)
```

Convert one input column to one “stack” in the output

Processed output for the first YearMonth

Remaining YearMonths to process

Stack all the processed months together

M

Column name for the unpivoted attribute

The data as a table

```
Table.UnpivotOtherColumns(df, {"Item"}, "YearMonth", "Value")
```

ID variables (not unpivoted and repeated in each stack)

Column name for the values from the unpivoted attribute

Python

The data as a
Pandas dataframe

Column name for the
unpivoted attribute

```
df.melt("Item", var_name = "YearMonth")
```

ID variables (not
unpivoted, repeated in
each stack)

R

The data as a dataframe

Column name for the unpivoted attribute

```r
melt(df, variable.name = 'YearMonth', id = 'Item')
```

ID variables (not unpivoted, repeated in each stack)