IterationInsights

Data Engineering Execution Optimization

Jarid McKenzie

- Partner & Lead Architect
- 10+ Years of BI Experience
- Teacher, Mentor & Community Leader

IterationInsights

Who I actually am!

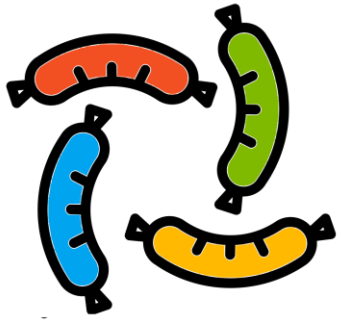
- My Website
- LinkedIn
- YouTube channel coming soon
- GitHub (Foundatum)

What are we going to talk about

1. Why use an Execution Framework?
2. What do these typically look like?
3. How do we optimize this?

Why use an Execution Framework?

What are we even doing?



Request

Deliver

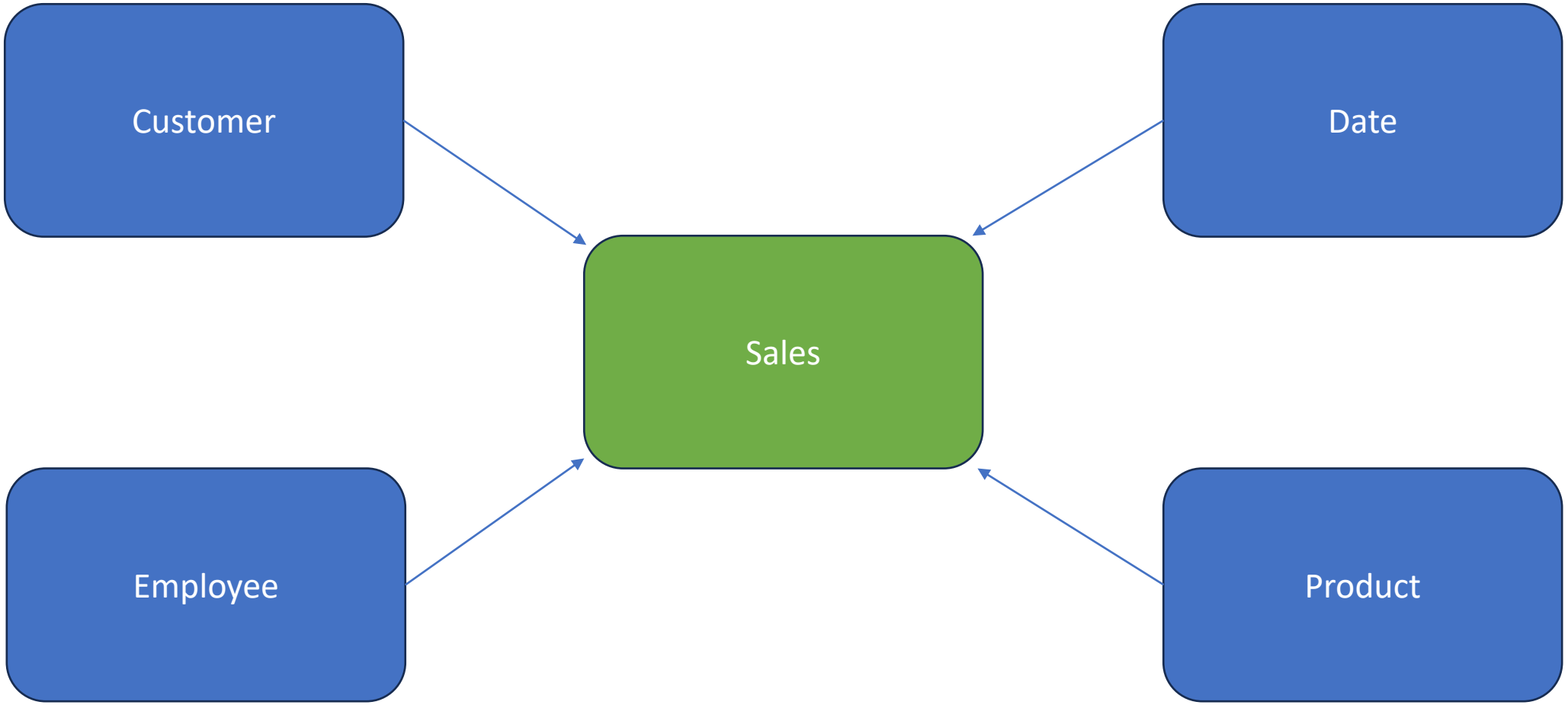
Pay

Request

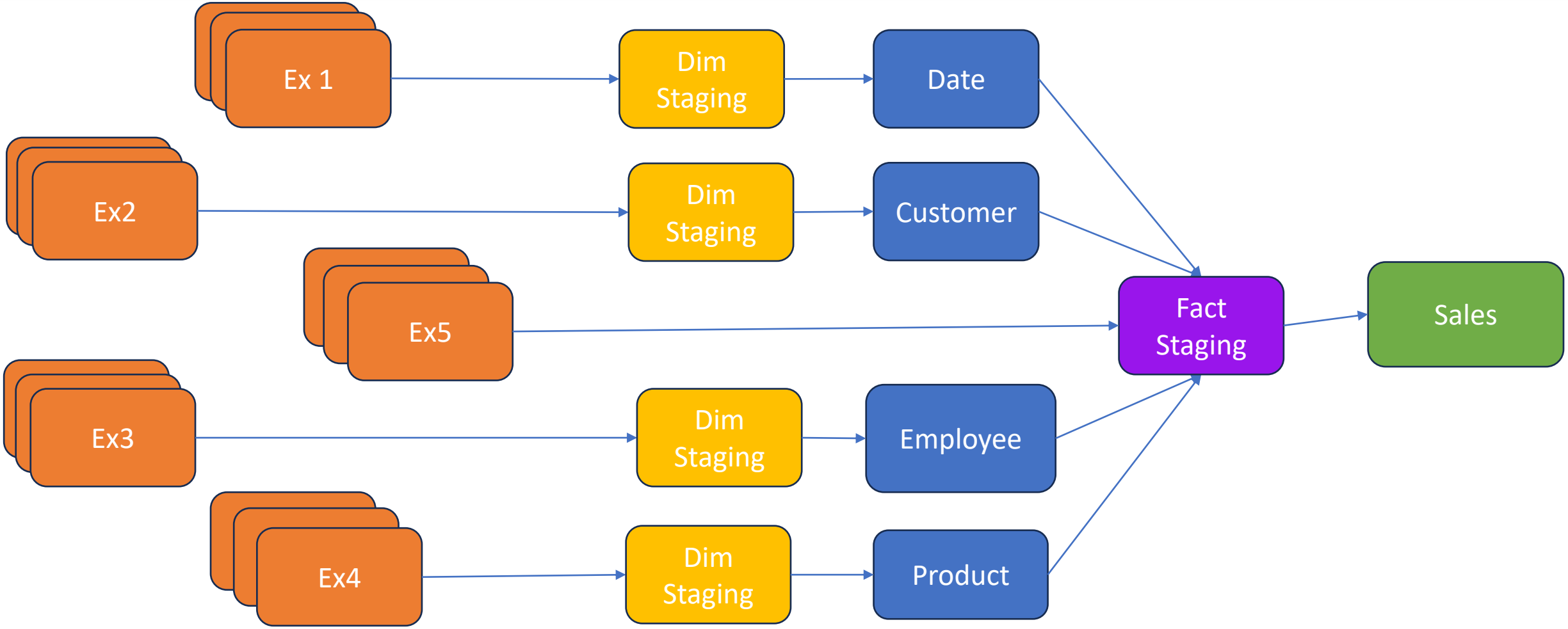
Deliver

Pay

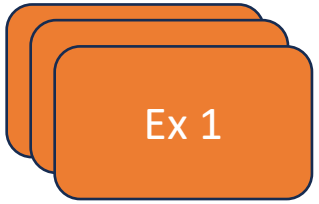
What are we even doing?



What are we even doing?



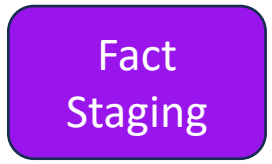
Common Patterns in Data Engineering



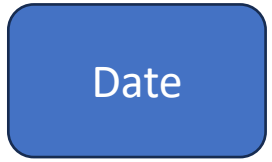
Extracts – Full/Incremental

Integrate into ODS

Data Staging



Replace Surrogate Keys



Dimension Loads (SCD 1 & 2)

Incremental Fact Load

Why use a Framework

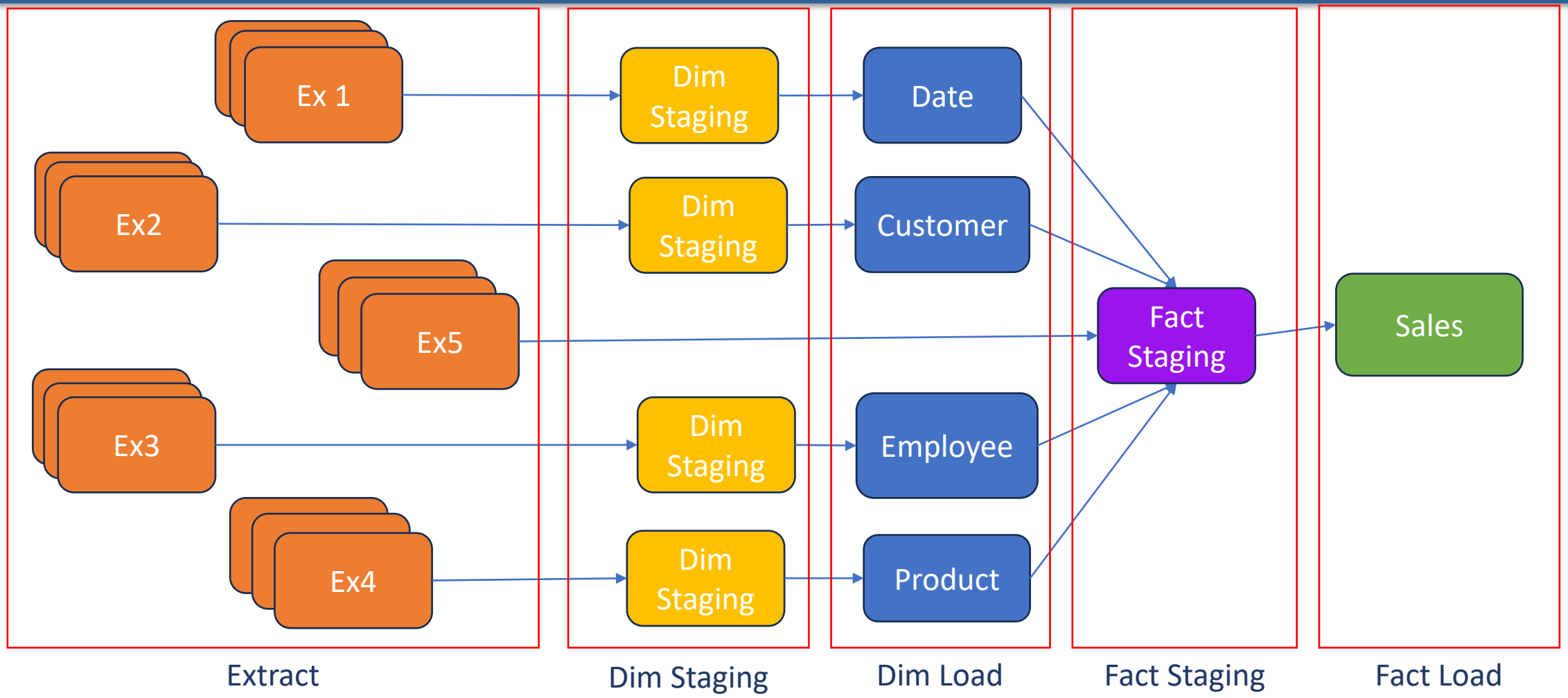
Frameworks help us to **standardize** and **organize** our execution

Pro Tip #1

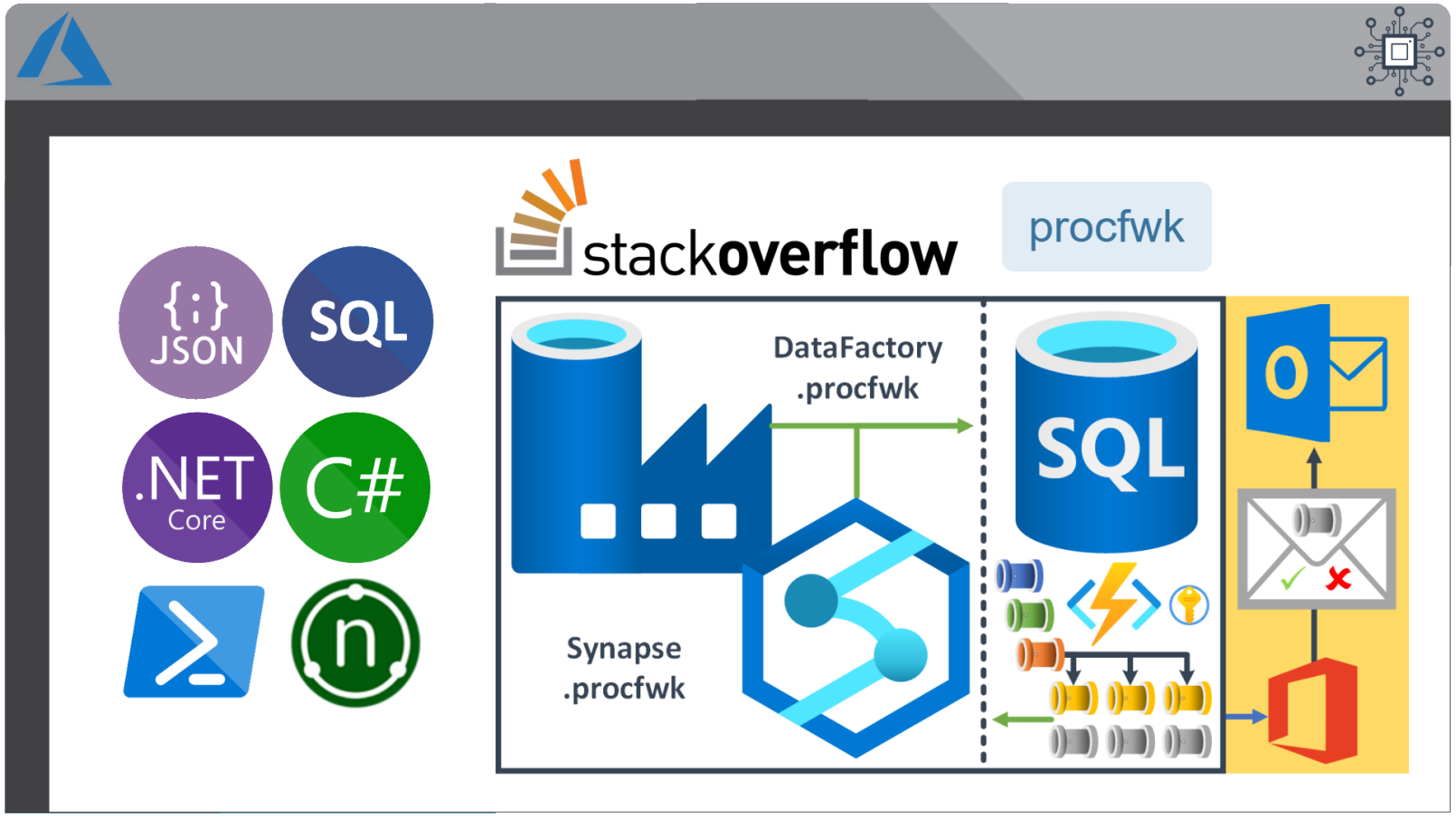
If it ain’t broke,
don’t fix it!

What do Frameworks Look Like?

Using Stages



Wonderful Open-Source Framework

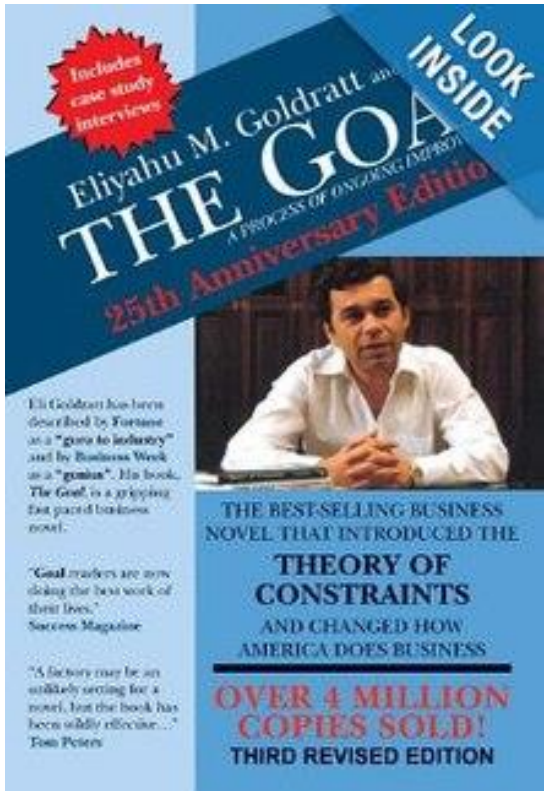


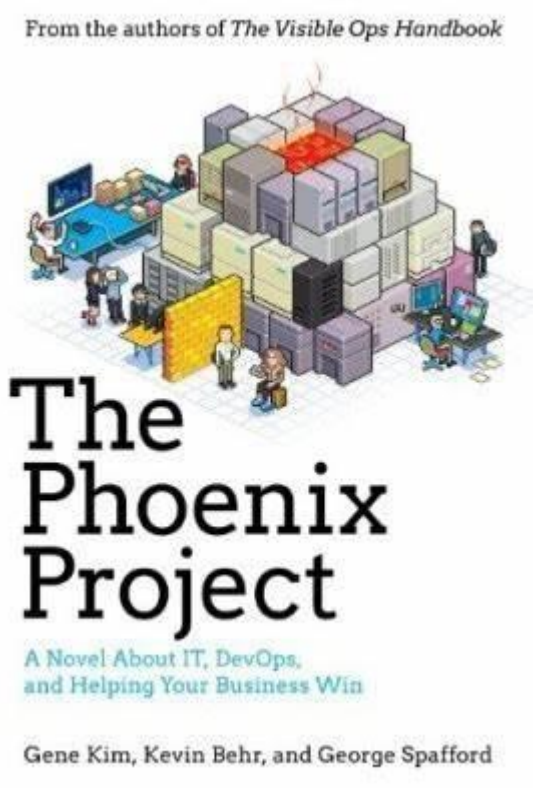
Just Search for:

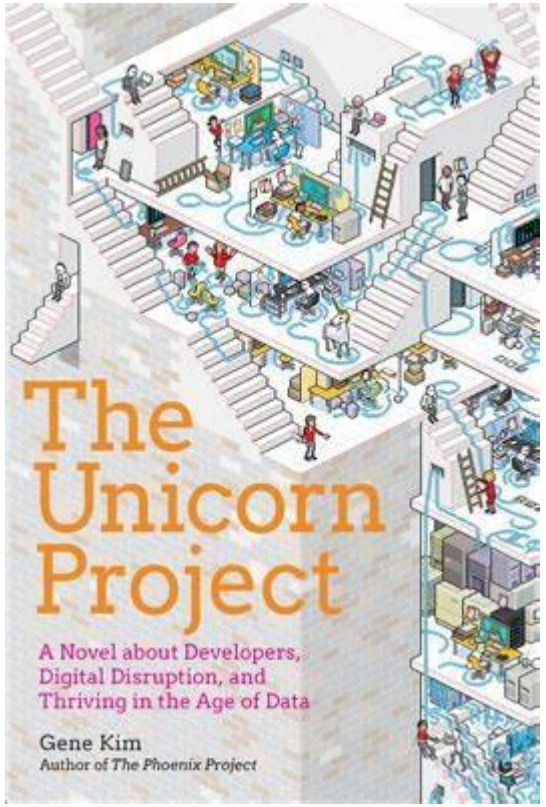
PROCFWK

All credit goes to
Paul Andrew

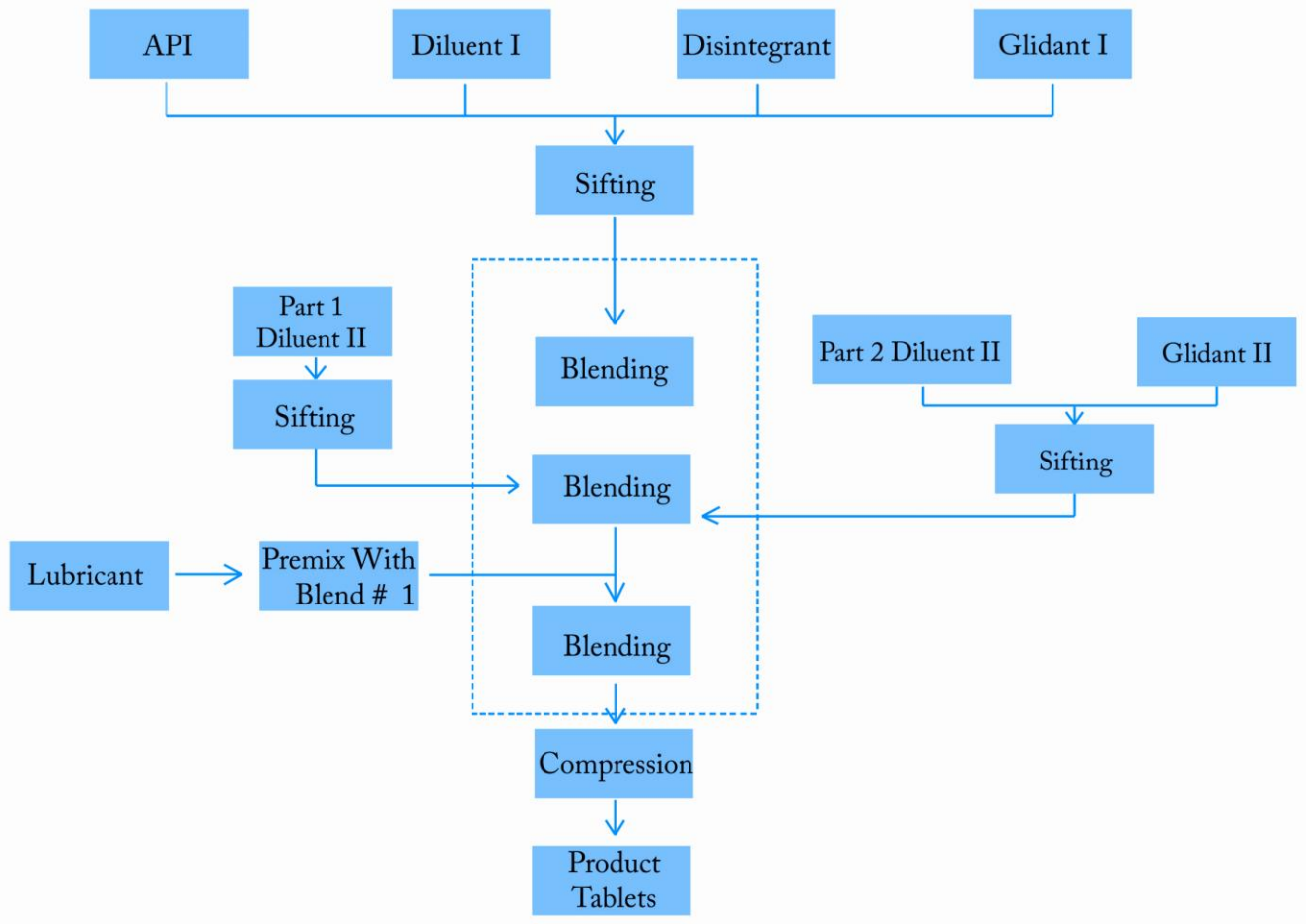
Books that motivated this approach







Analogy from Manufacturing

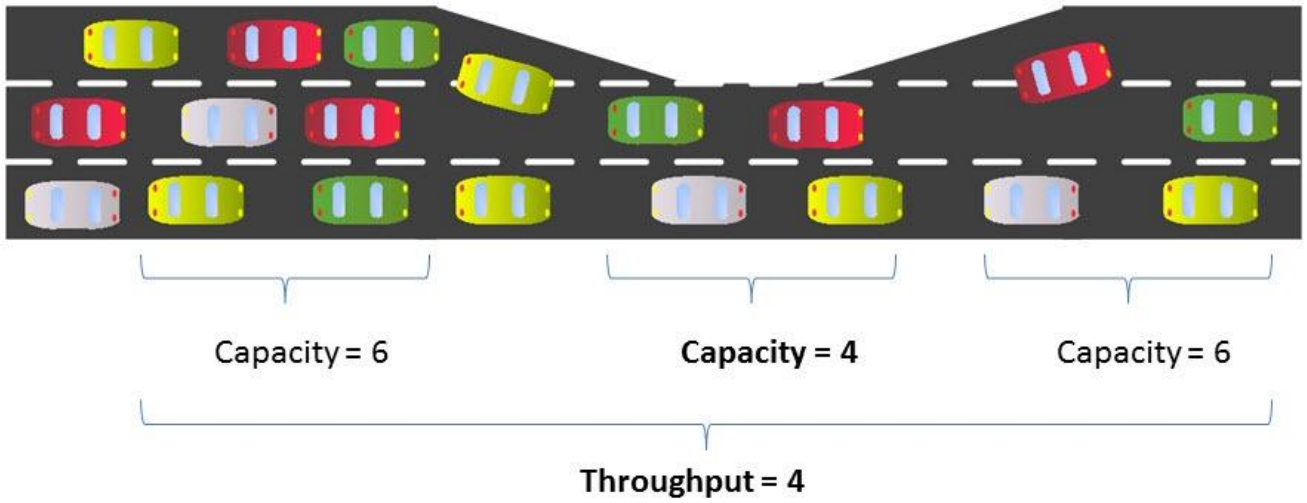


In Manufacturing, there are many steps that need to be taken to arrive at a finished product.

Some can be done in parallel, some in series. Most need to be performed using **separate, specialized** equipment.

The key takeaway is to **Identify the Bottlenecks**.

Bottlenecks Data Engineering (Synapse)



Identify the Bottlenecks:

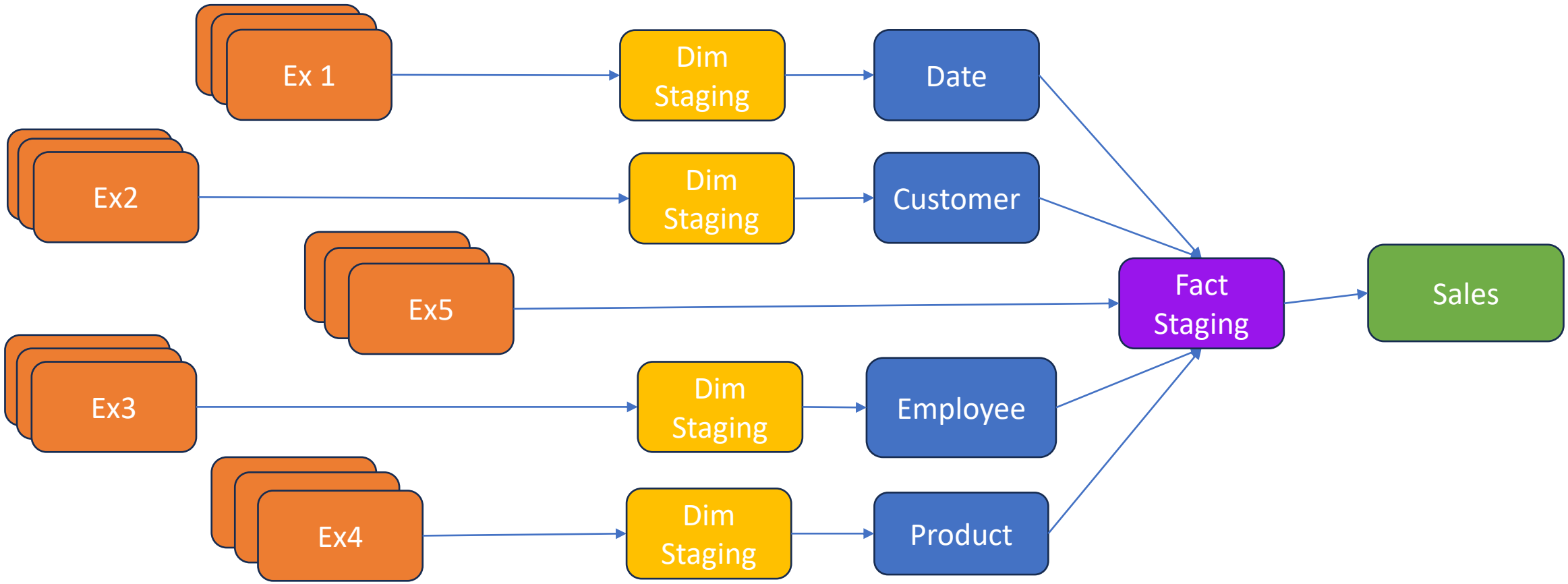
- Self-hosted Integration Runtimes
- Servers that we're pulling data from
- Rate limited APIs
- Spark Pools

Remember that this is cloud processing. We can run significantly more operations in parallel than a physical manufacturing plant.

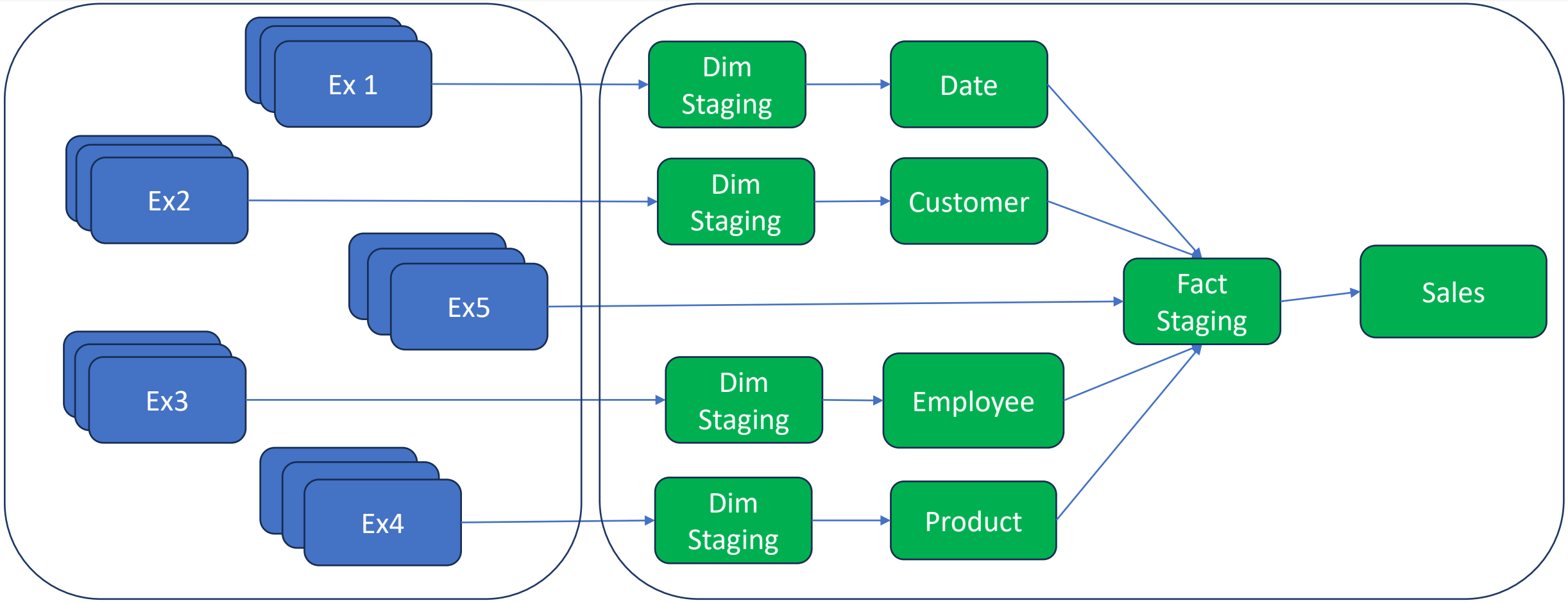
Pro Tip #2

Make your tech stack as narrow as necessary

Back to the task flow



Back to the task flow

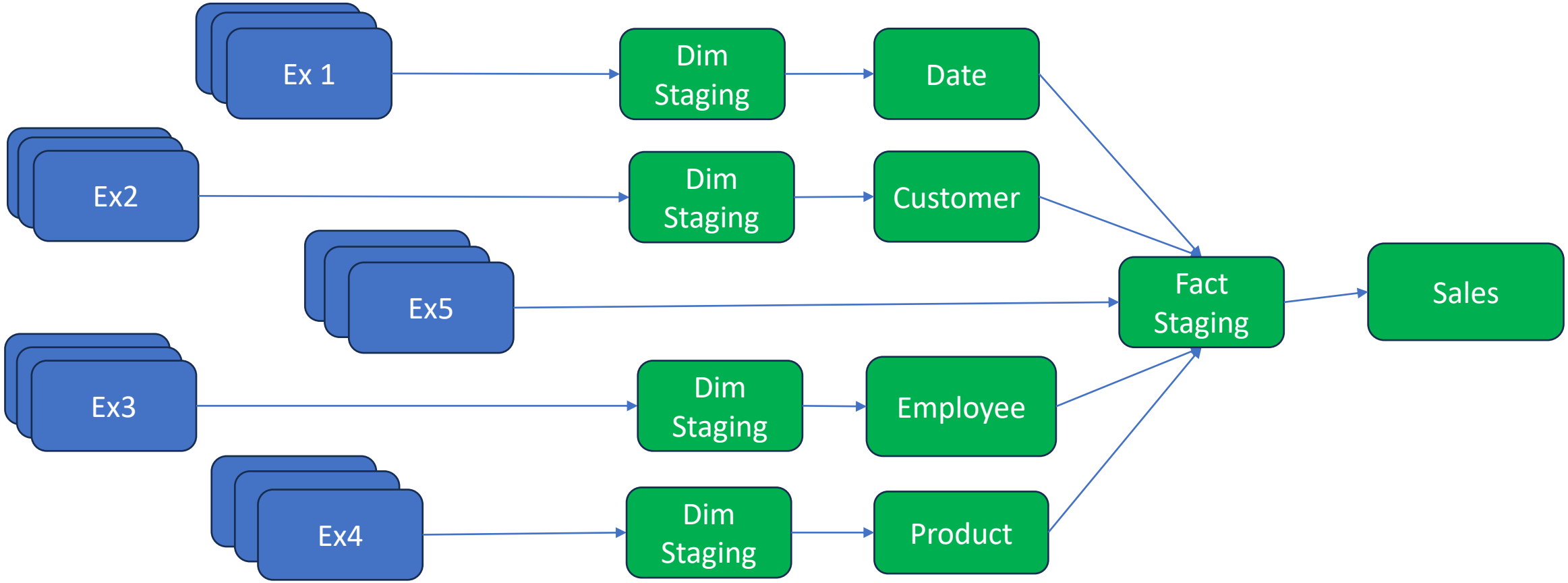


Copy Data

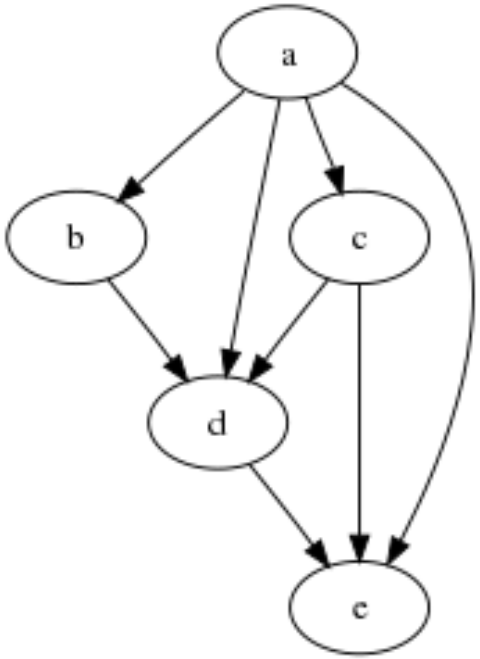
Spark

How Do We Optimize?

What the heck is a DAG?



What the heck is a DAG?



Directed – the edges within a graph have direction from one vertex to another

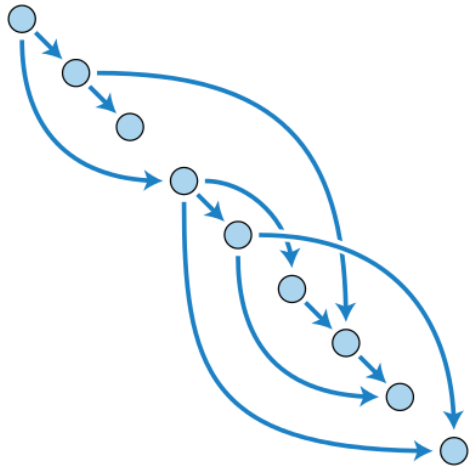
Acyclic – the graph contains no cycles. Once a vertex has been visited, there is no way to ‘walk’ back to that vertex

Graph – A set of vertices (objects) and edges (relationships). An edge joins two vertices

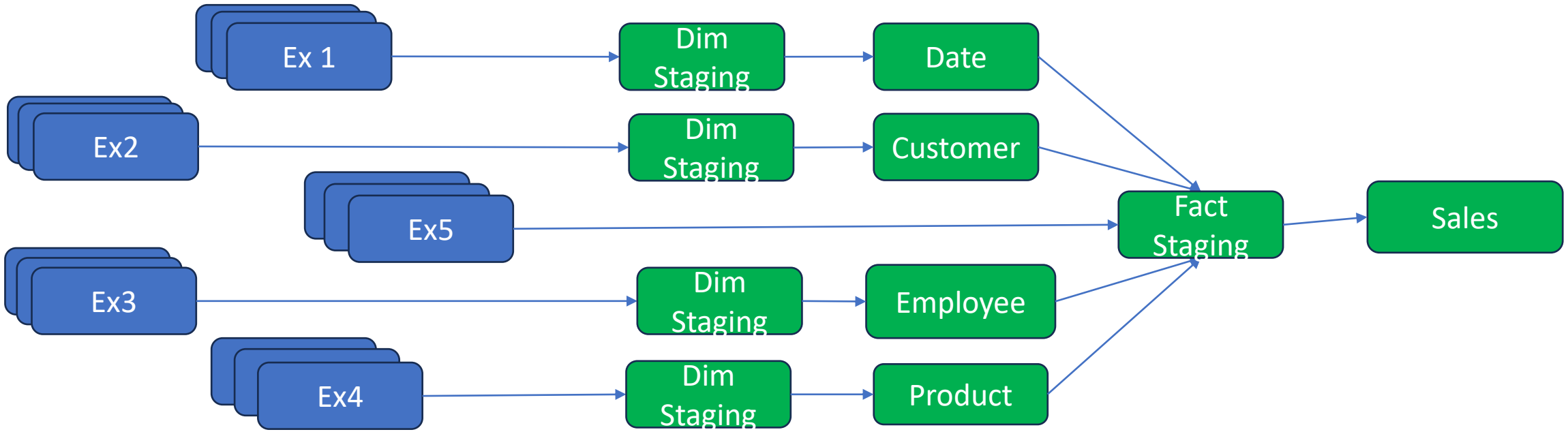
DAGs Help Us Schedule

Topological Sorting:

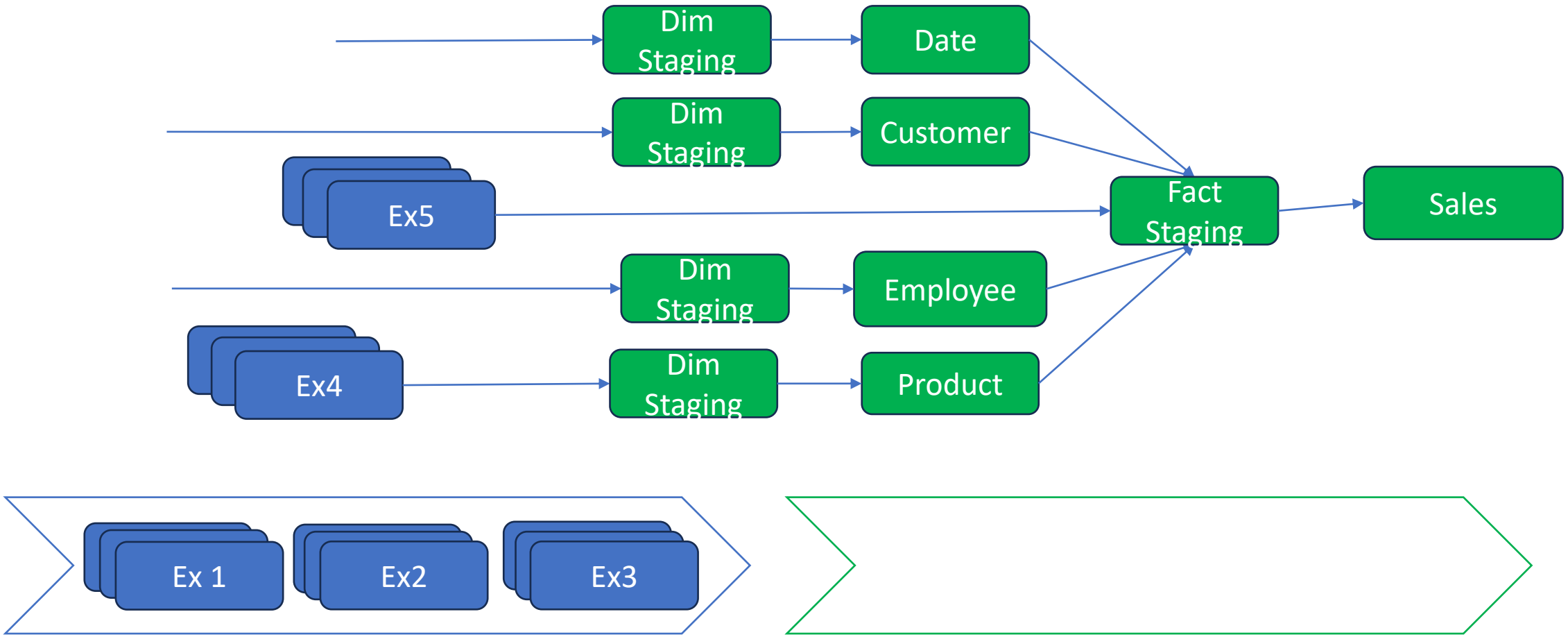
Arranging the nodes of the graph in such a way that we can complete them one after the other.



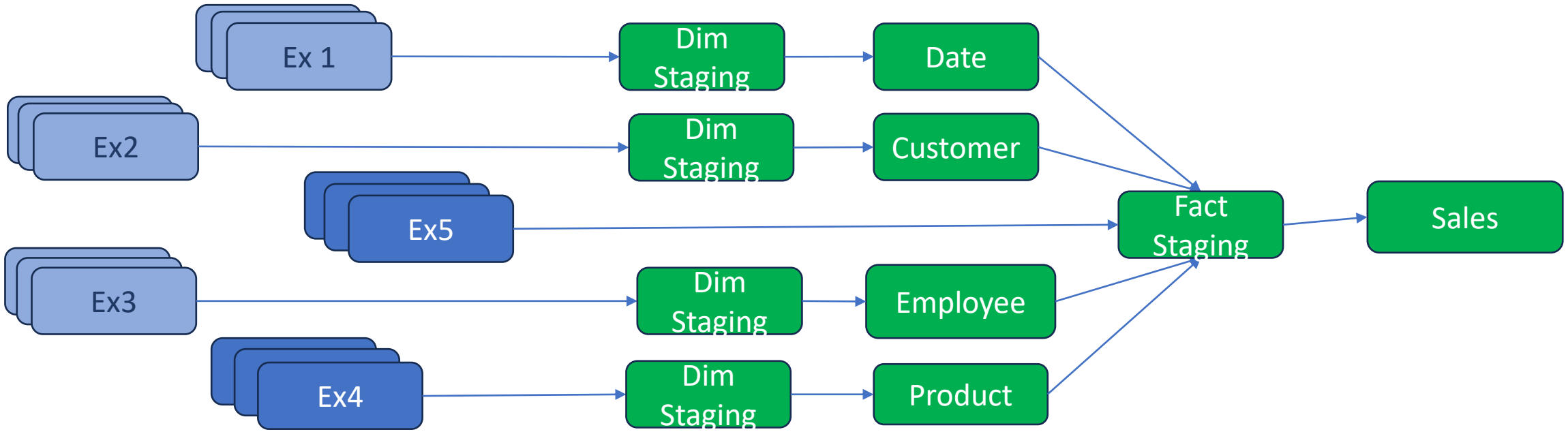
Create a Queue for each Bottleneck



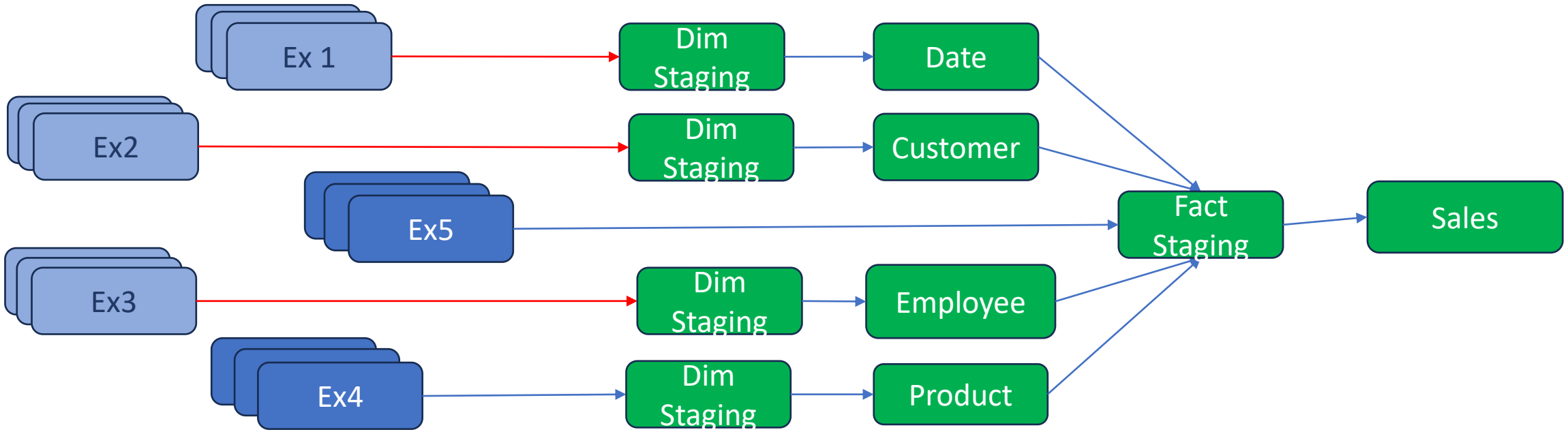
Create a Queue for each Bottleneck



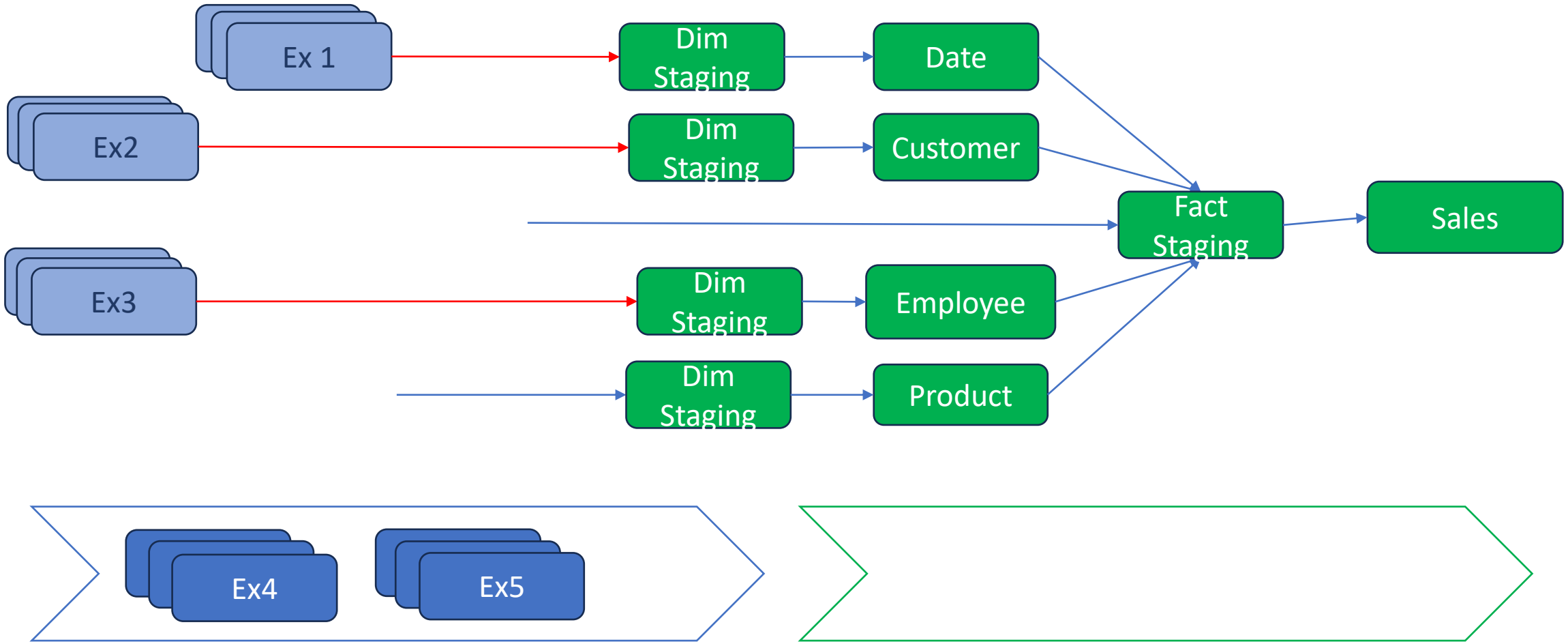
Create a Queue for each Bottleneck



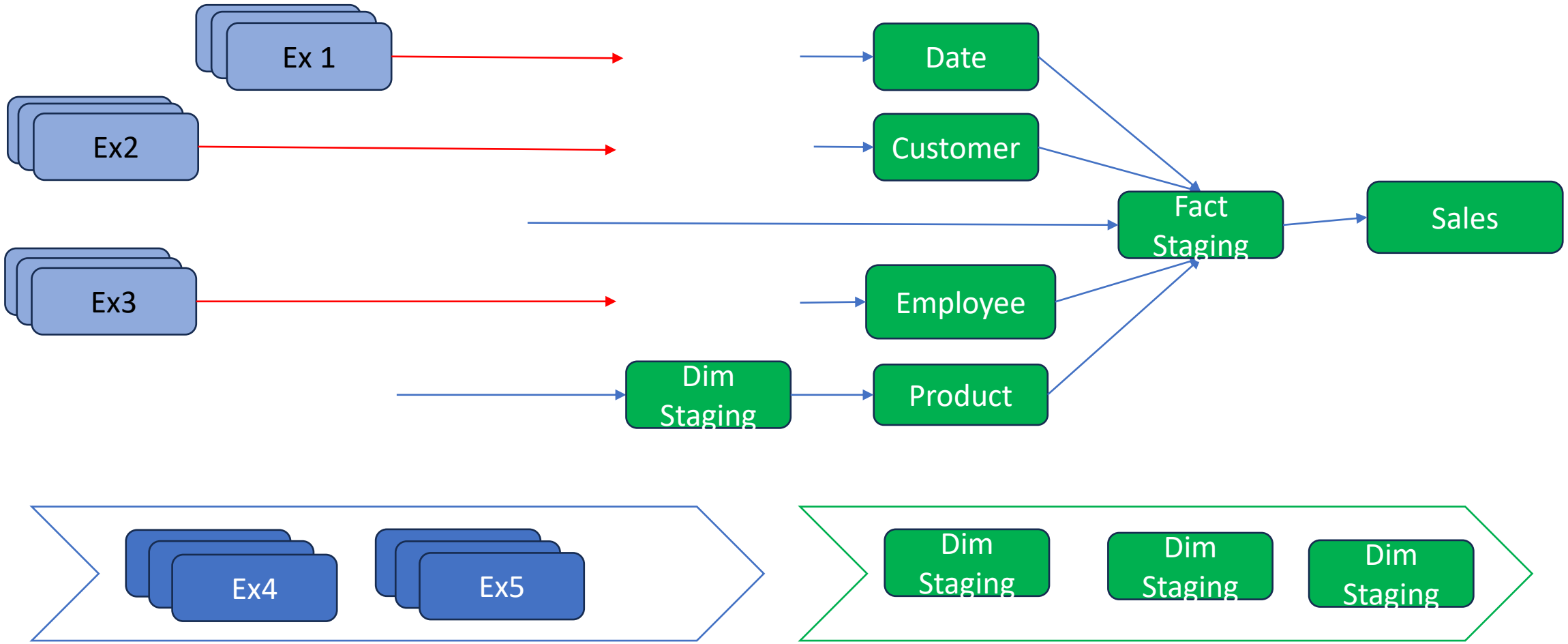
Create a Queue for each Bottleneck



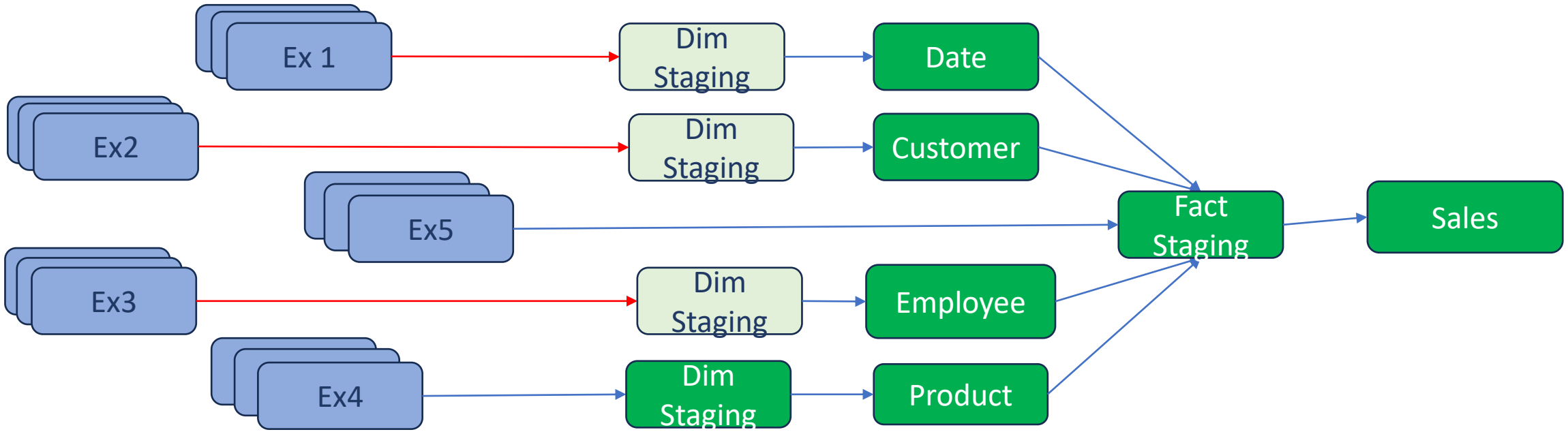
Create a Queue for each Bottleneck



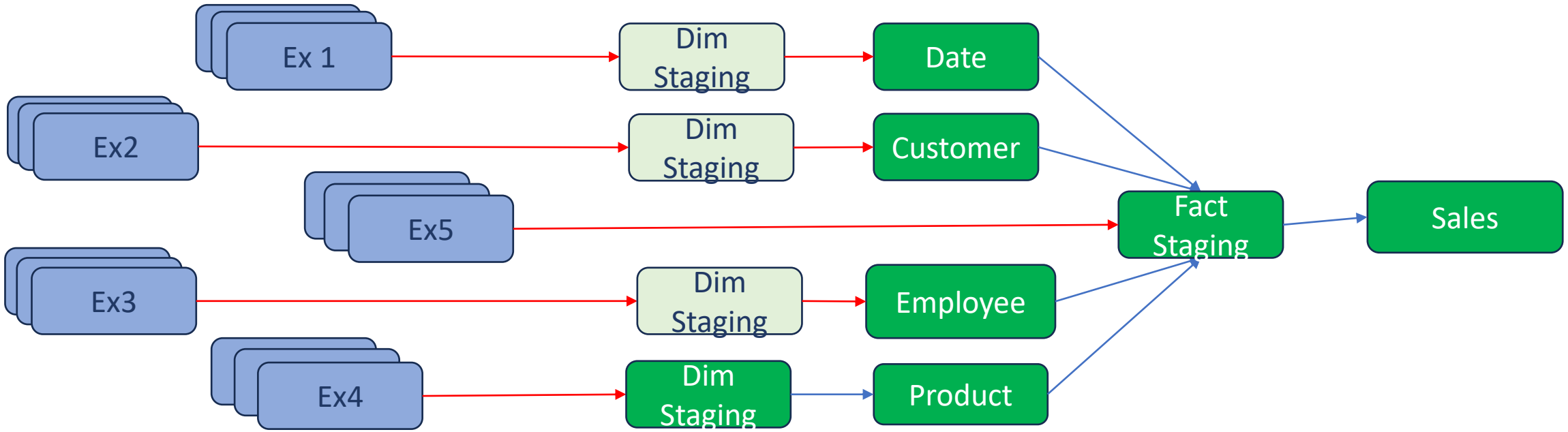
Create a Queue for each Bottleneck



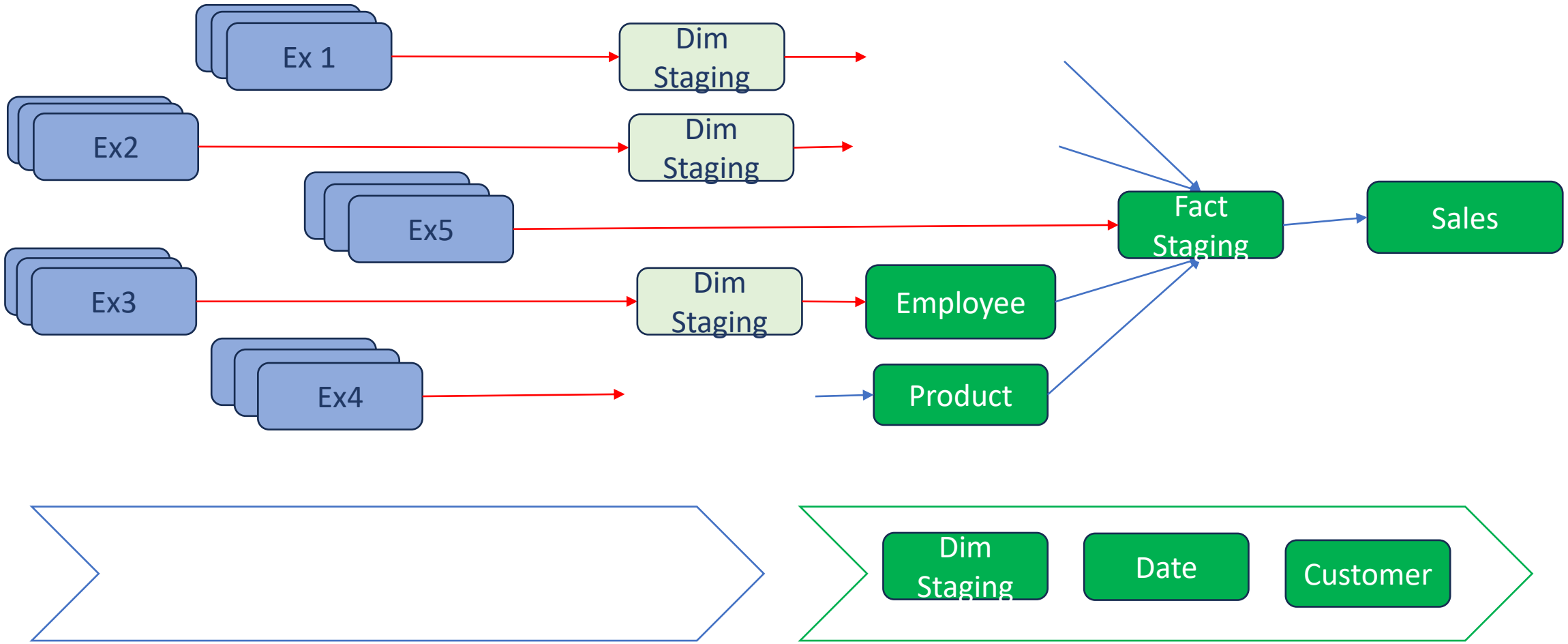
Create a Queue for each Bottleneck



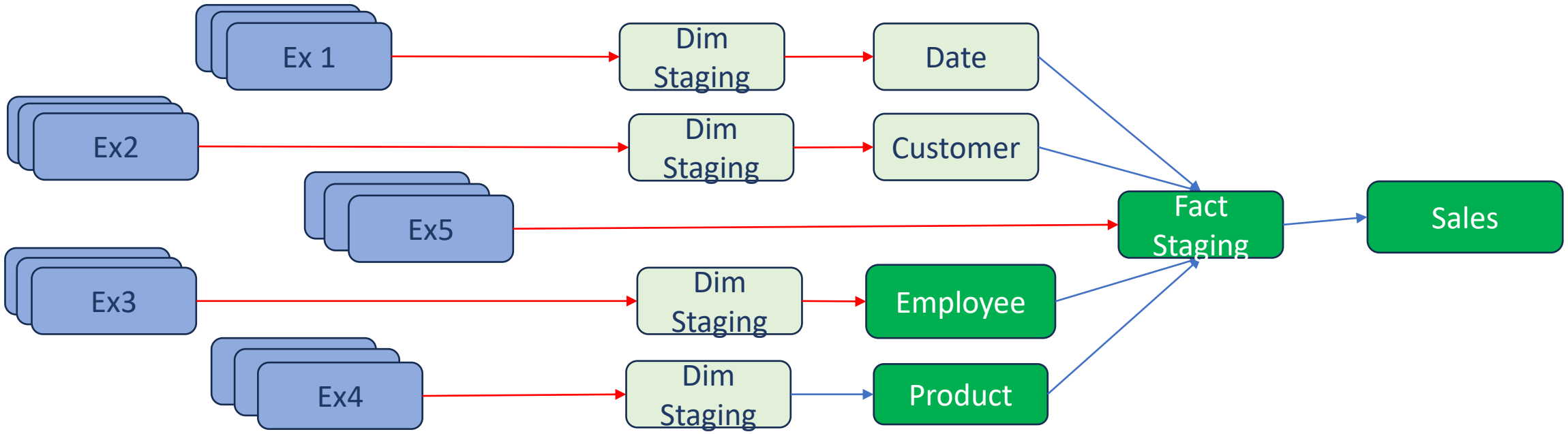
Create a Queue for each Bottleneck



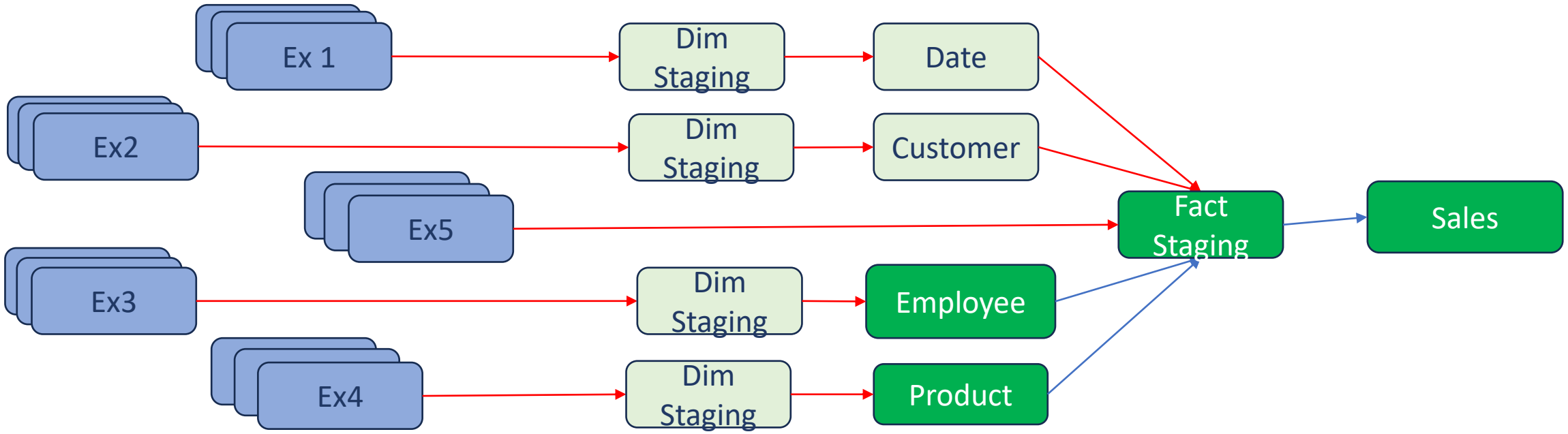
Create a Queue for each Bottleneck



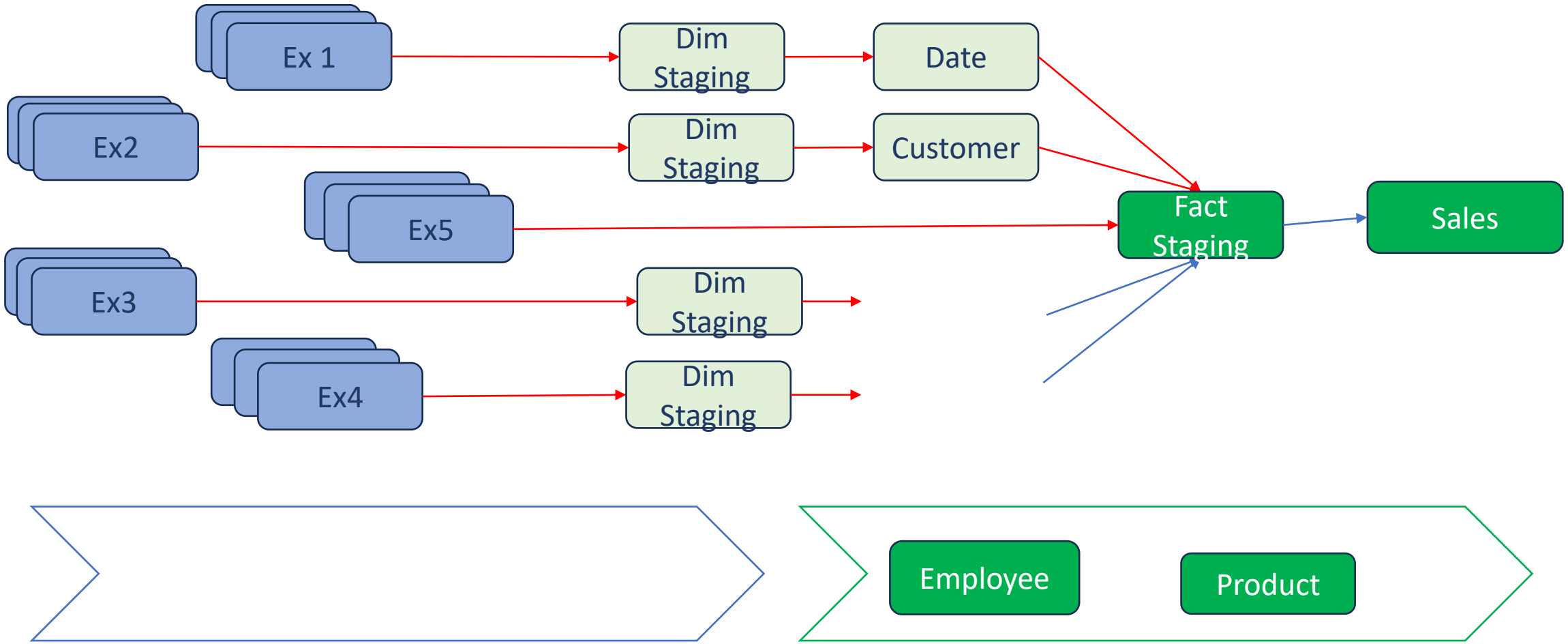
Create a Queue for each Bottleneck



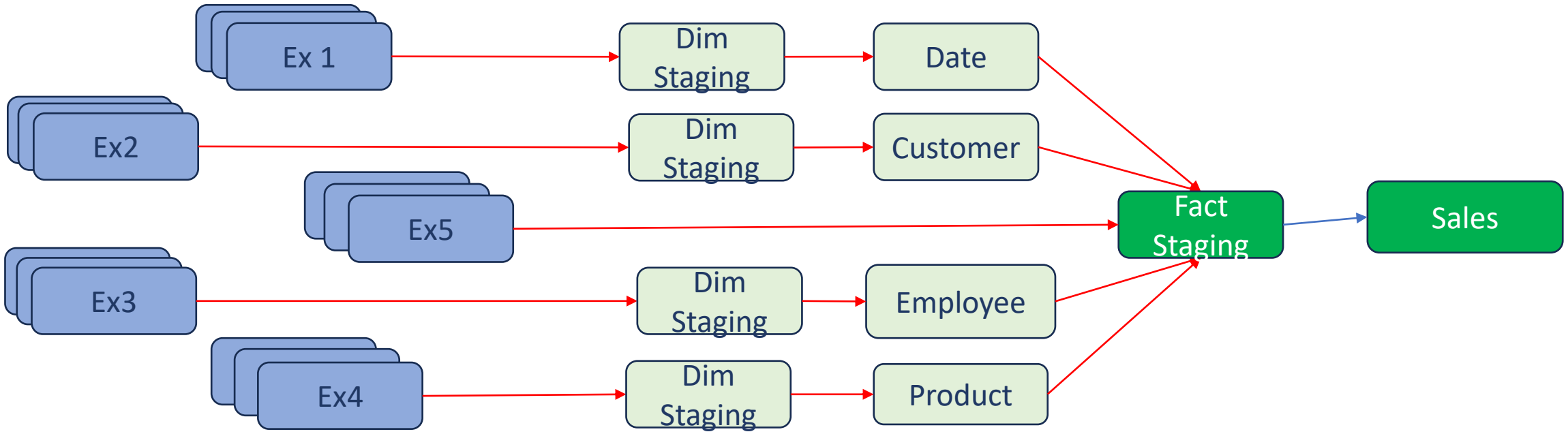
Create a Queue for each Bottleneck



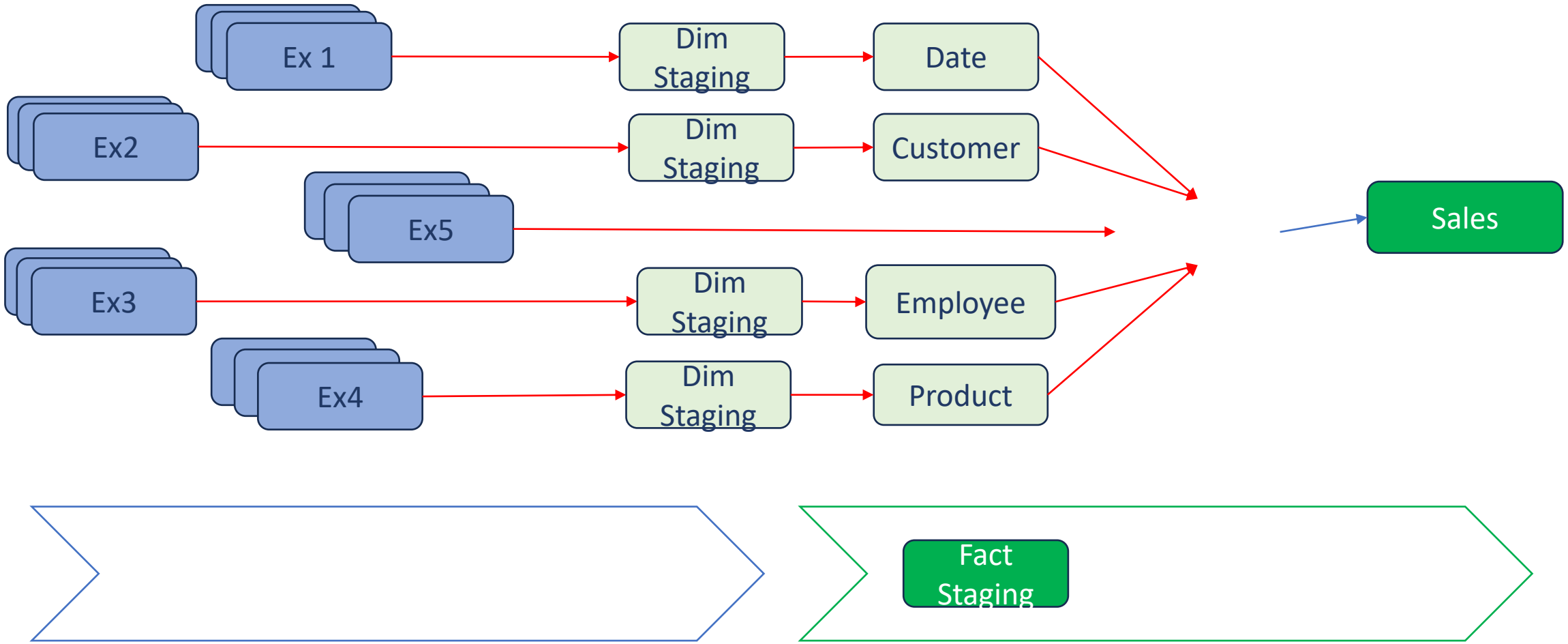
Create a Queue for each Bottleneck



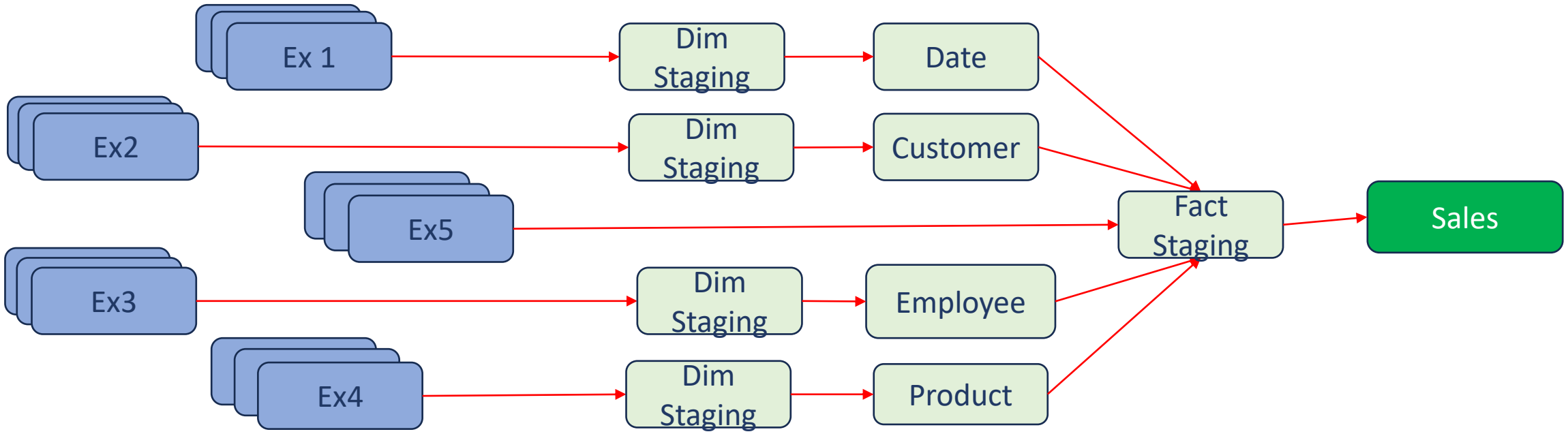
Create a Queue for each Bottleneck



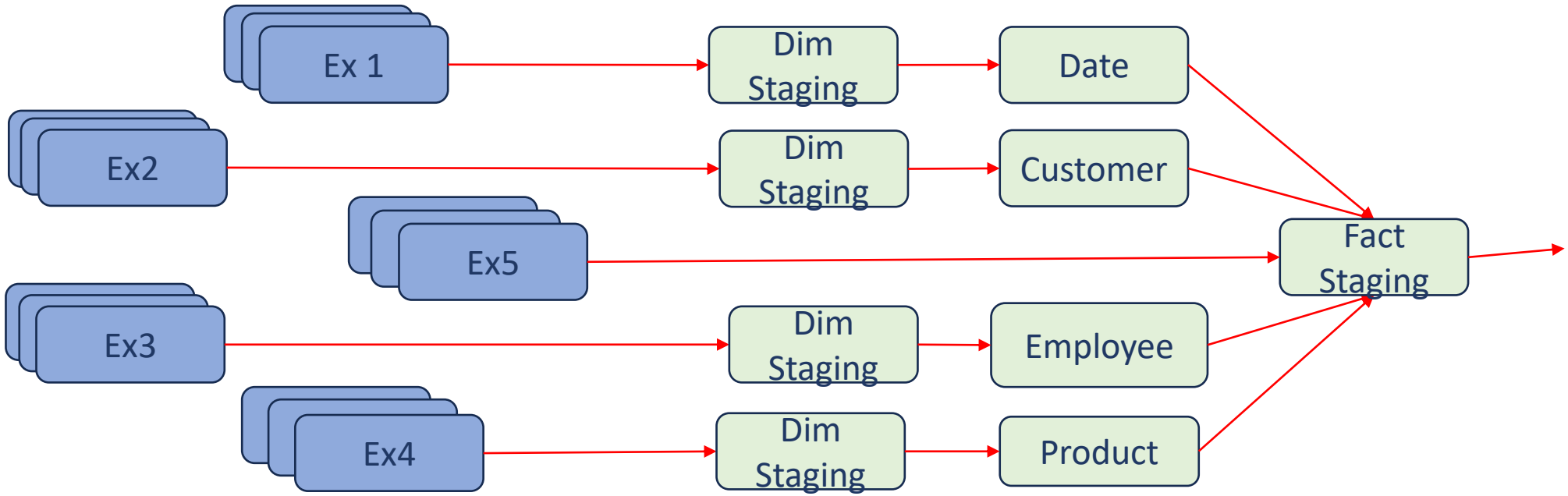
Create a Queue for each Bottleneck

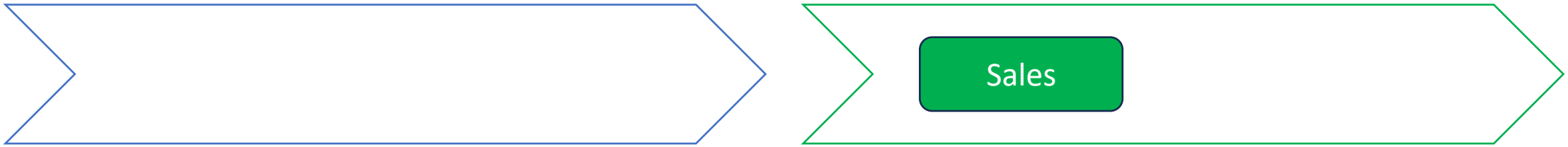


Create a Queue for each Bottleneck

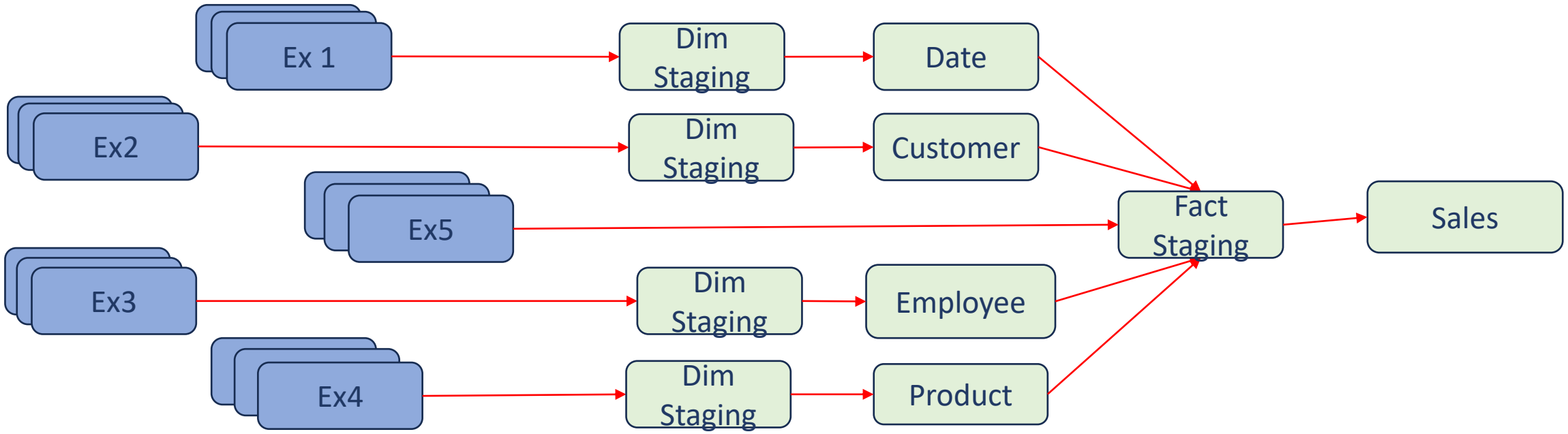


Create a Queue for each Bottleneck

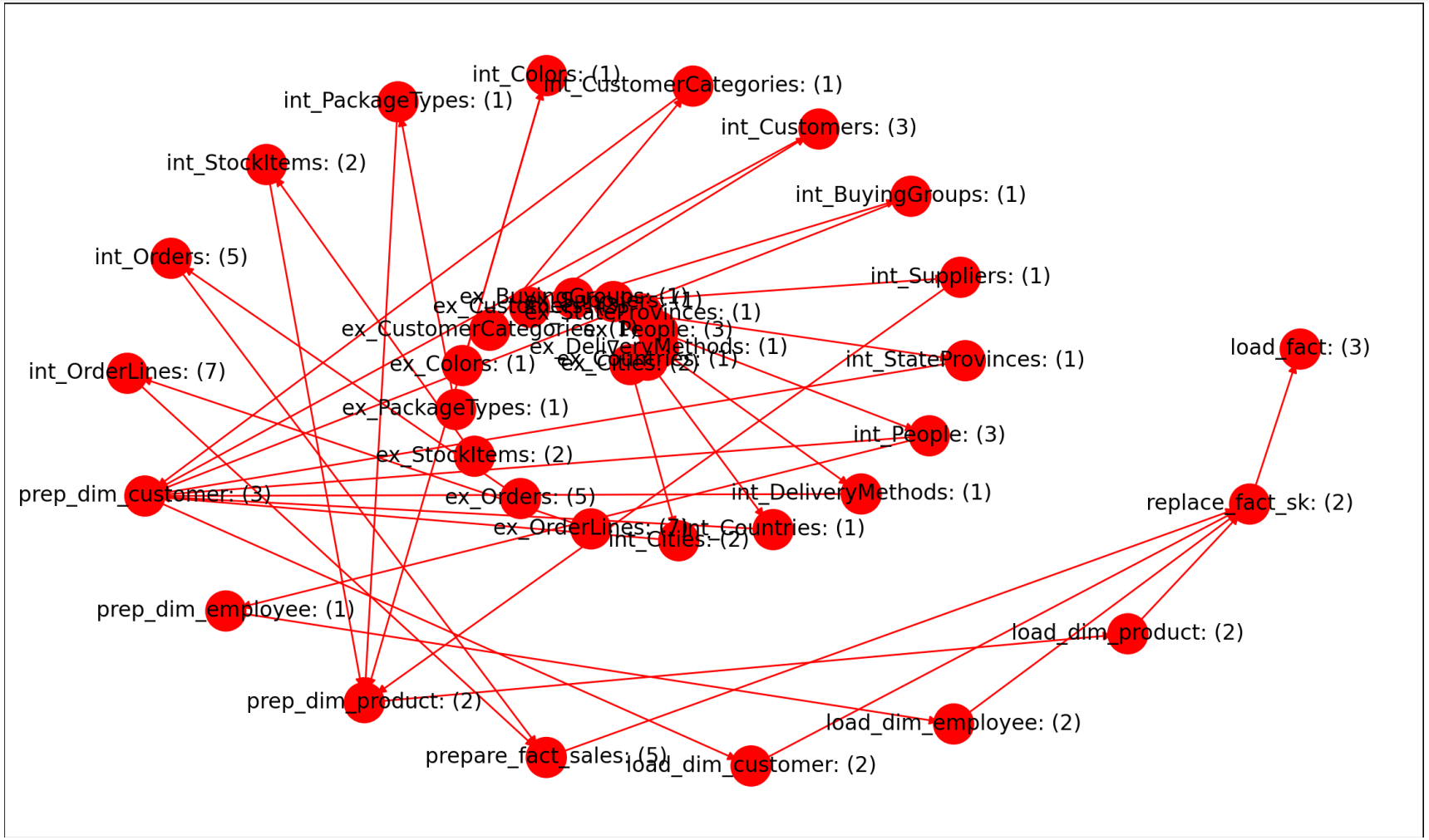




Create a Queue for each Bottleneck



Let’s see if the demo Works



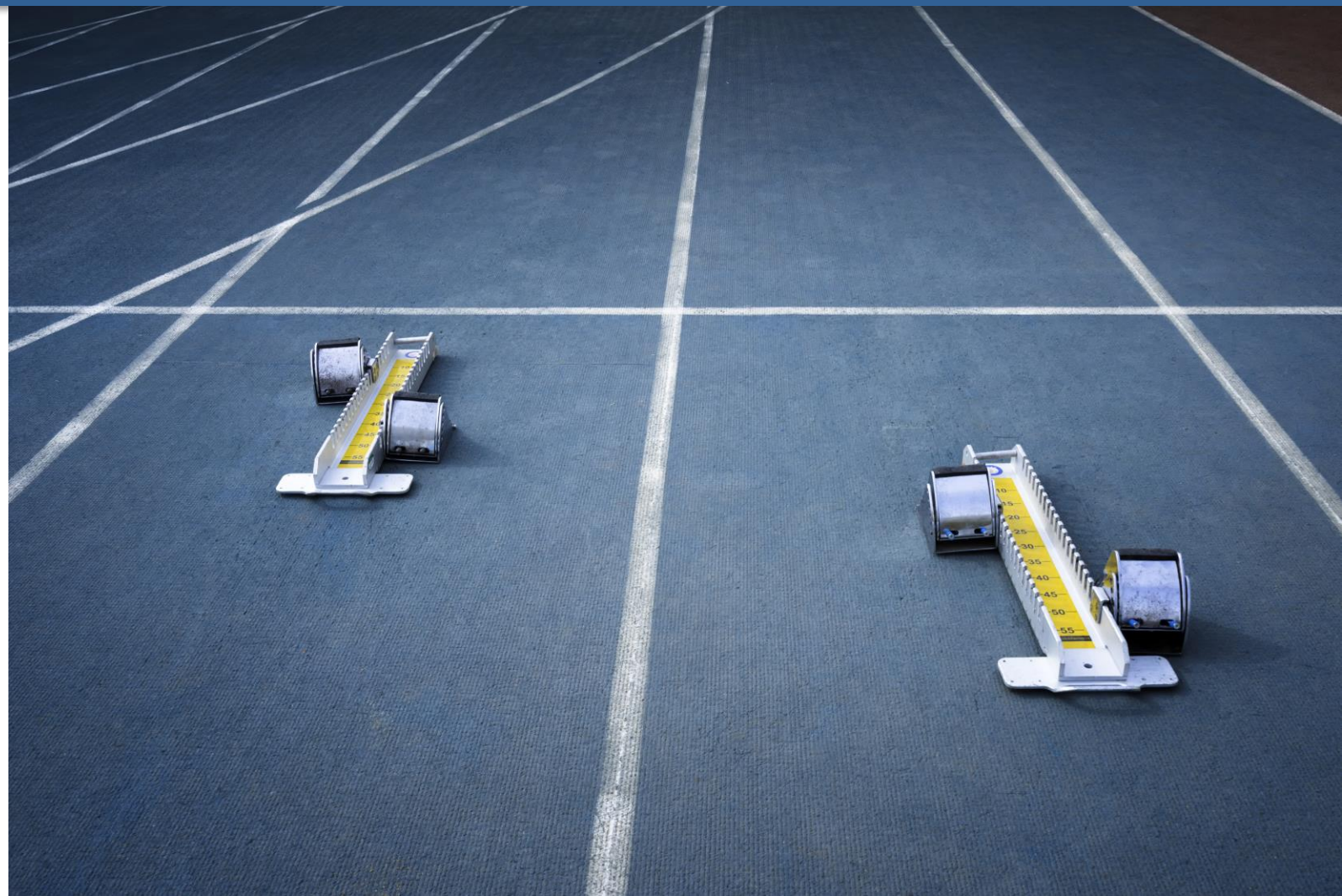
BUT WAIT!!!

Couldn’t we run anything that’s in the Queue?

How do we order the queue?

Easy Queue Ordering Strategies

1. Longest Waiting (FIFO)
2. Shortest Average Runtime
3. Longest Average Runtime



Longest Waiting (FIFO)

Include the timestamp of when the task is added to the queue.

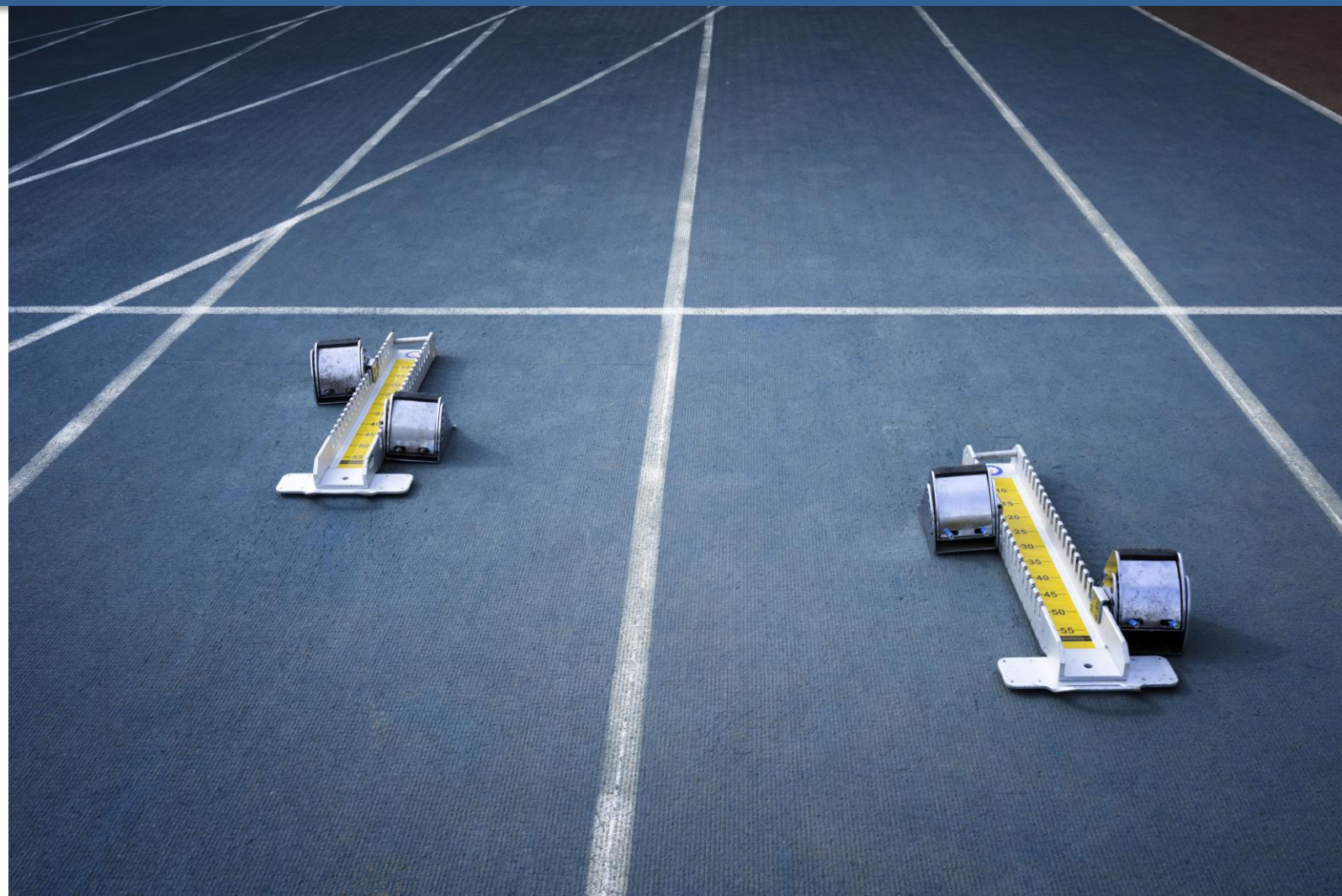
Shortest and Longest Average Runtime

When you run tasks, record both the start and end timestamps.

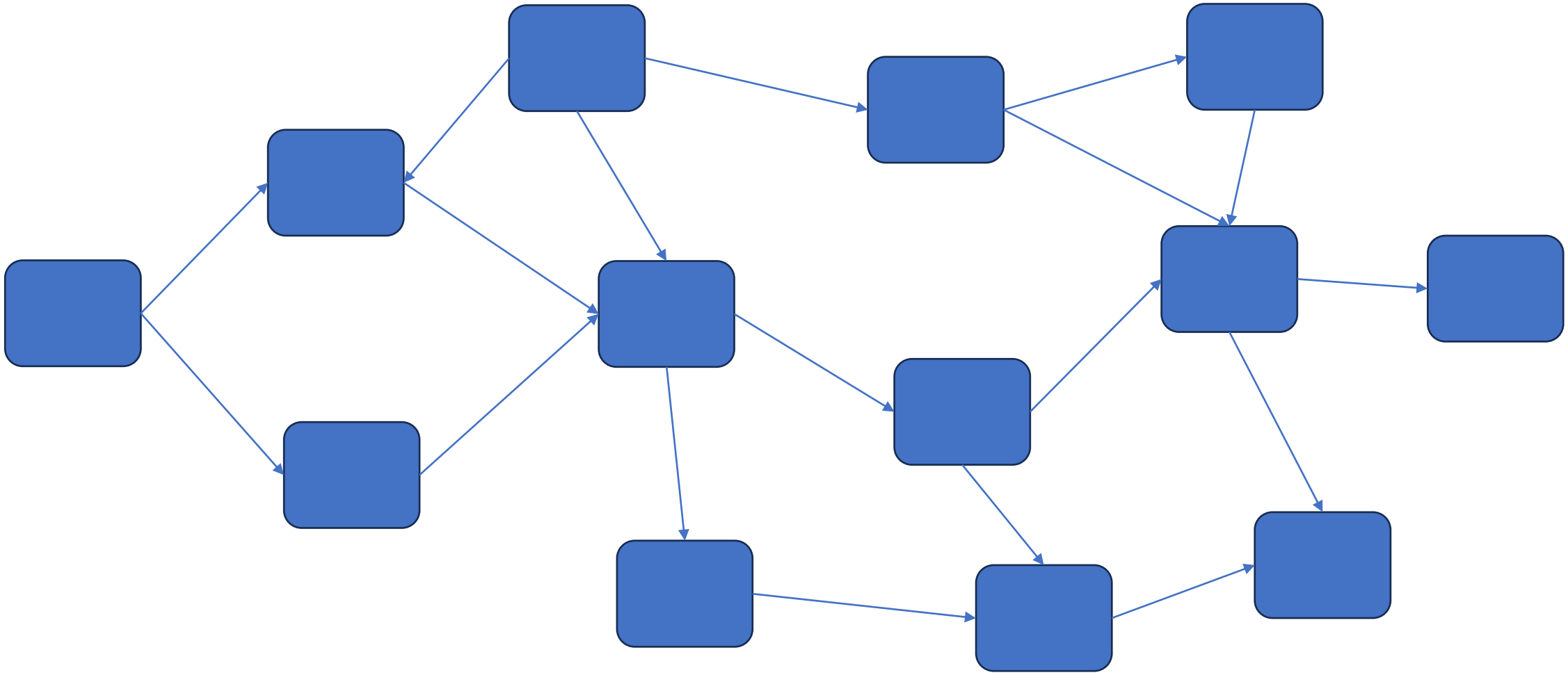
Take a recent sample of package runs and average when you create the DAG

Hard Queue Ordering Strategies

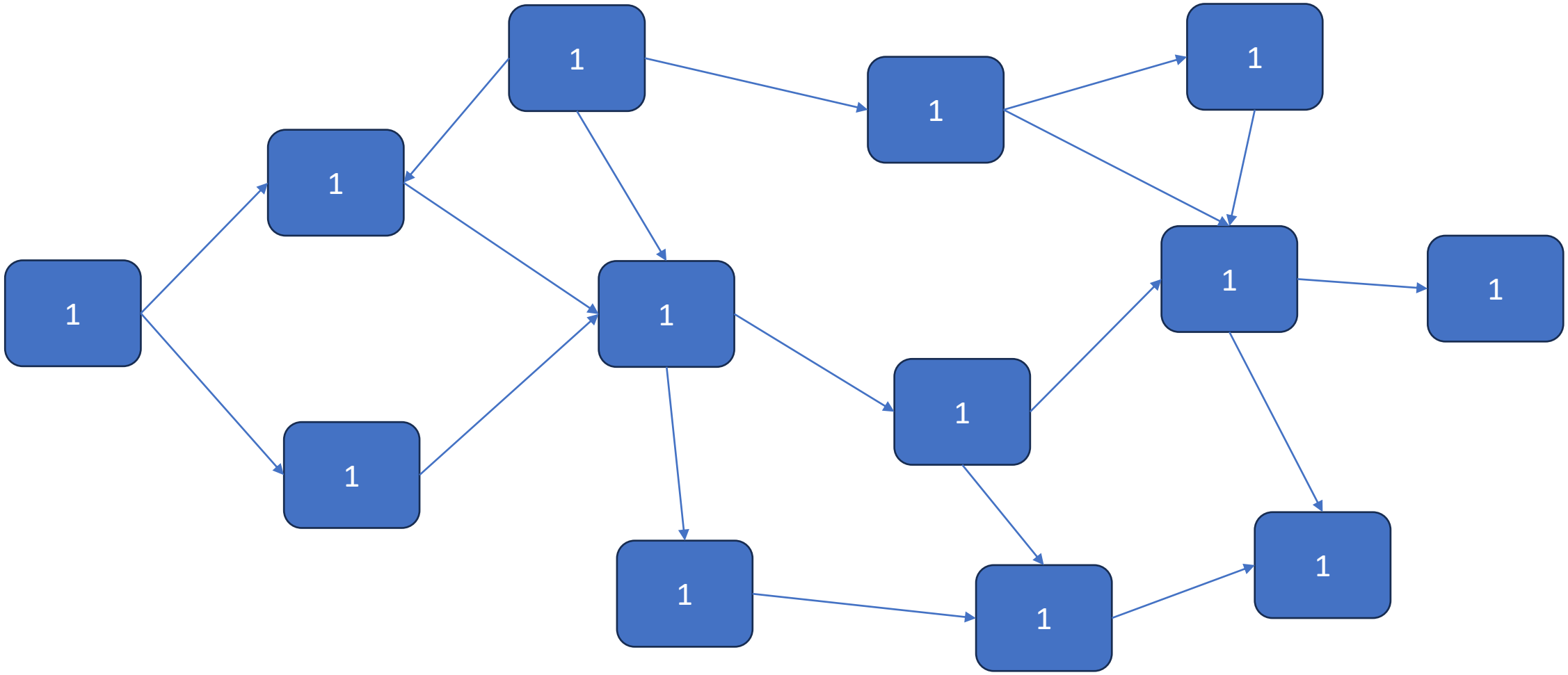
1. Most Dependent Tasks
2. Longest Cumulative Dependent Tasks



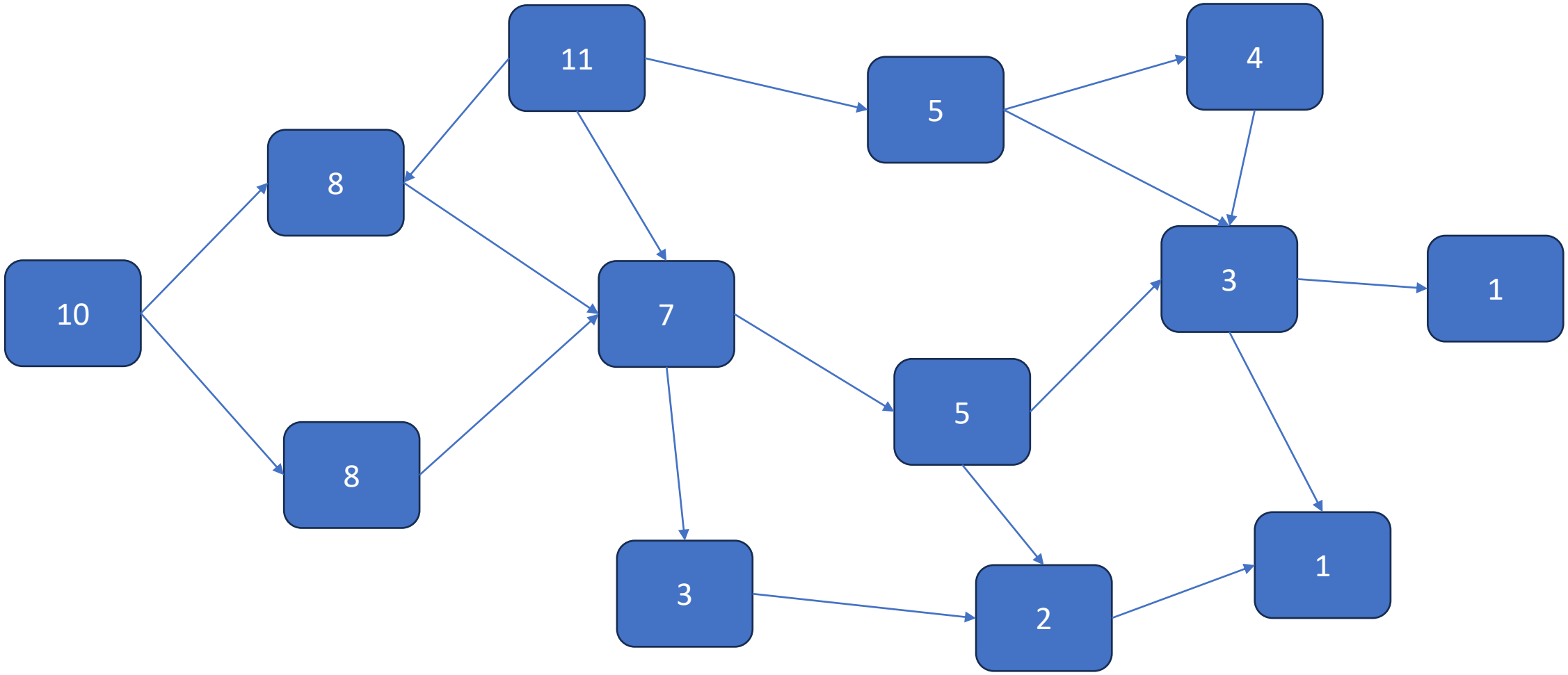
Dependent Tasks



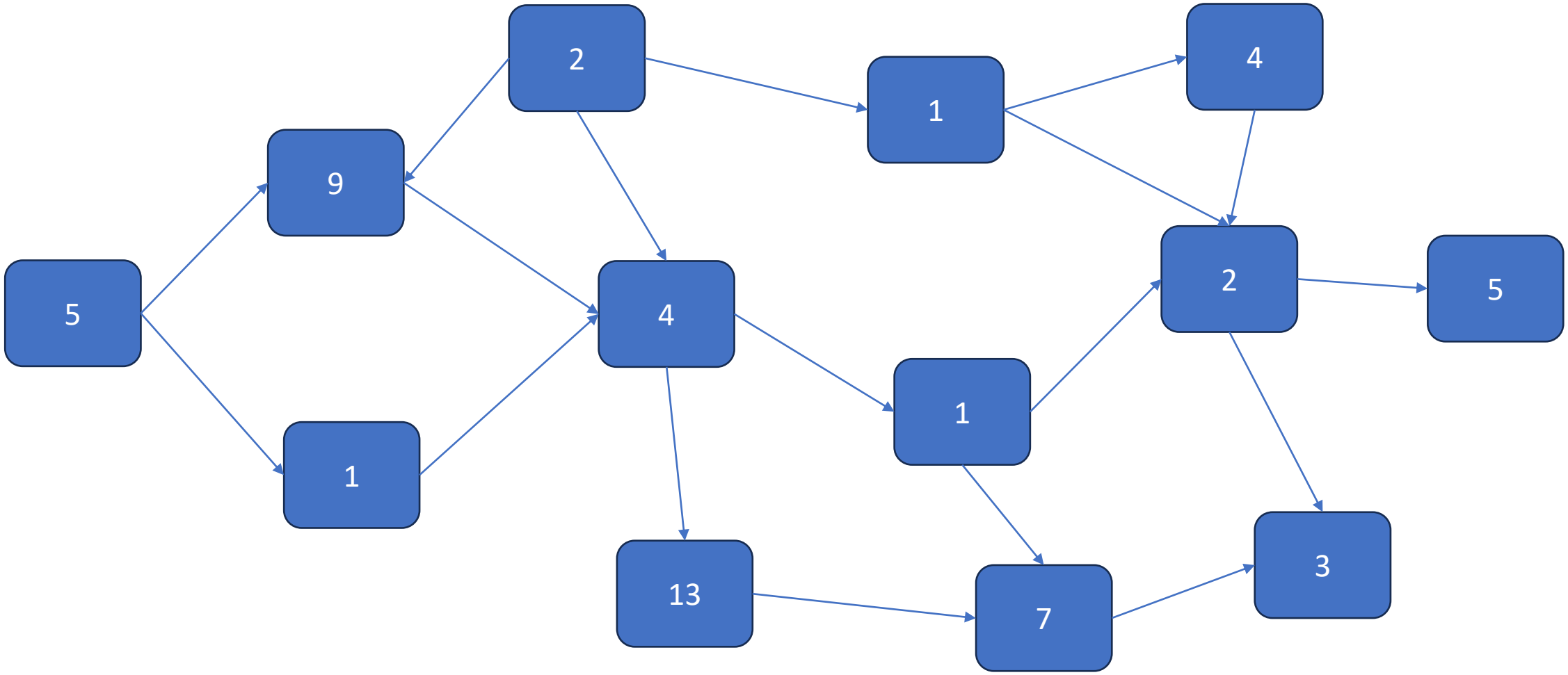
Dependent Task Calculation



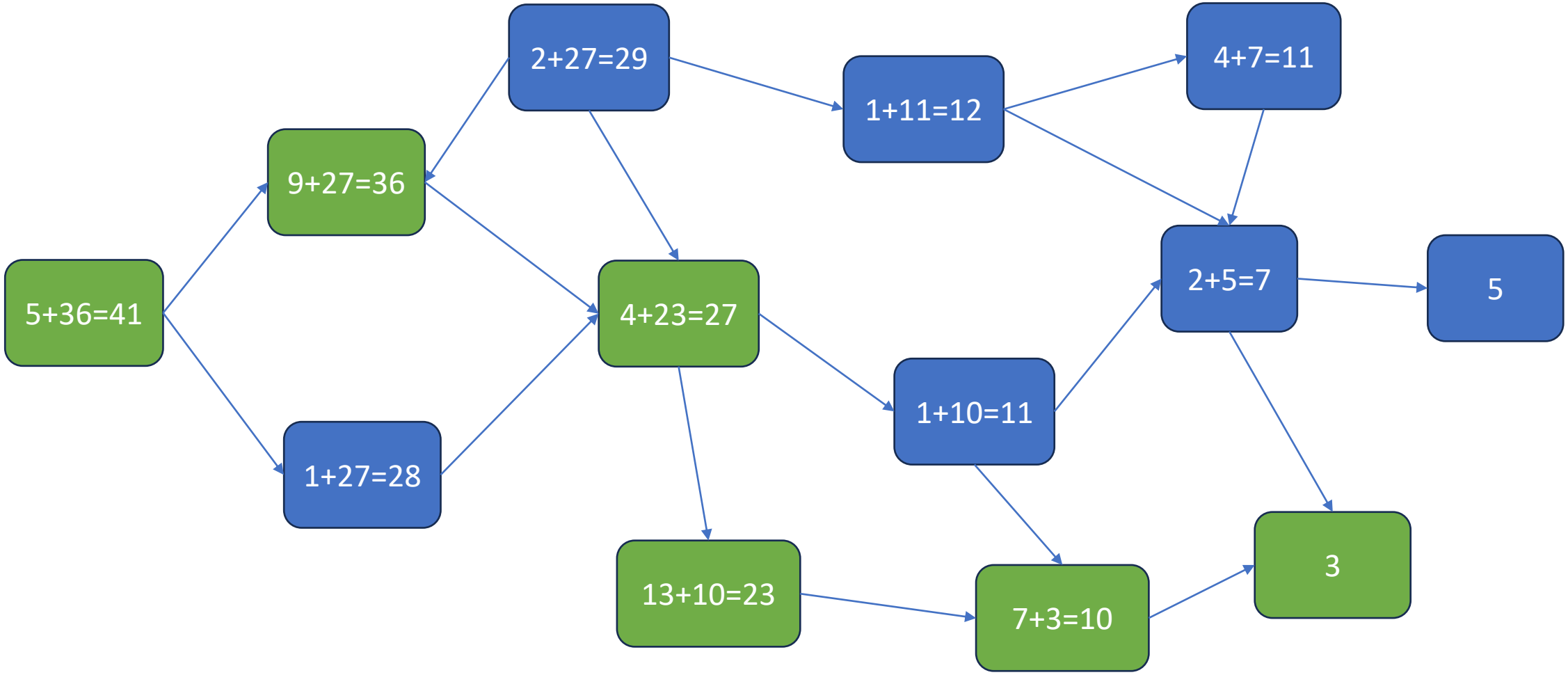
Dependent Task Calculation – DFS or BFS



Longest Dependency – Finding the Critical Path



Longest Dependency – Finding the Critical Path



Who I actually am!

- My Website
- LinkedIn
- YouTube channel coming soon
- GitHub (Foundatum)