**Got It!**

## 1. Scheduling data

Great job on these exercises! Now that we understand data processing, let's talk about scheduling.

## 2. Scheduling

Scheduling can apply to any task we listed in the previous data processing lesson. In this lesson though, to demonstrate scheduling, we will focus on updating tables and databases to keep things straightforward and easy to understand. Scheduling is the glue of a data engineering system. It holds each small piece and organizes how they work together, by running tasks in a specific order and resolving all dependencies correctly.

## 3. Manual, time and sensor scheduling

There are different ways to glue things together. For example, we can run tasks manually. If an employee is moving from the United States to Belgium, and therefore changing offices, someone can request an immediate update and we can update the table right away ourselves. However, there are downsides with human dependencies. Ideally, we'd like our pipeline to be automated as much as possible. Automation is when you set some tasks to execute at a specific time or condition.

## 4. Data pipeline

For example,

## 5. Data pipeline

For example, we could update the employee database every morning at 6 AM. If a new employee was added the previous day, then the change will be reflected in the morning.

## 6. Manual, time and sensor scheduling

Or, we could set some tasks to execute if a specific condition is met. This is called sensor scheduling.

## 7. Data pipeline

For example,

## 8. Data pipeline

we could update the departments table only if a new employee was added to the employees table. There's really no reason to update otherwise.

## 9. Manual, time, and sensor scheduling

This sounds like the best option but it requires having sensor always listening to see if somethings been added. This requires more resources and may not be worth it in this case. Manual and automated systems can also also work together: if a user manually upgrades their subscription tier on the app, automated tasks need to propagate this information to other parts of the system, to unlock new features and update billing information.

## 10. Batches and streams

Another thing that matters is how the data is ingested. Data can be ingested in batches, which means it's sent by groups at specific intervals. Batch processing is often cheaper because you can schedule it when resources aren't being used elsewhere, typically overnight. For example, songs uploaded by artists may be batched and sent together to the databases every ten minutes, updates to the employees table can be batched every morning at 6:00 AM, and the revenue table used by the finance department can be updated overnight as well. The data can also be streamed, which means individual data records are sent through the pipeline as soon as they are updated. For example, if a user signs up, they want to be able to use the service right away, so we need to write their profile to the database immediately. Nowadays, it's inconceivable for a user to wait twenty-four hours to be able to use a service they just signed up for. Another example of batch vs. stream processing would be offline vs online listening. If a user listens online, Spotflix can stream parts of the song one after the other. If the user wants to save the song to listen offline, we need to batch all parts of the song together so they can save it. There's a third option called real-time, used for example in fraud detection, but for the sake of simplification and because streaming is almost always real-time, we will consider them to be the same in this course.

## 11. Scheduling tools

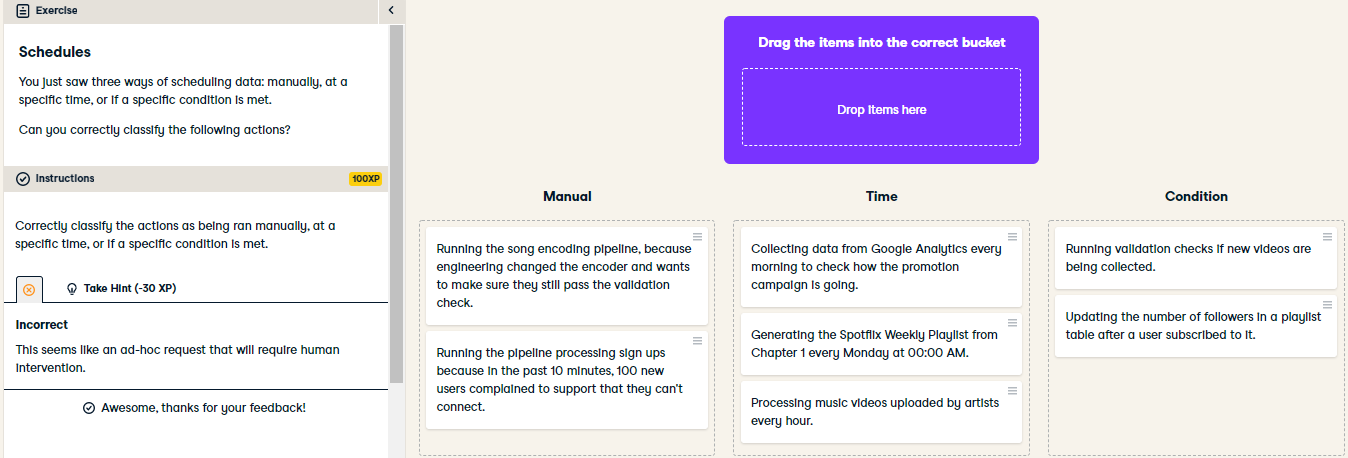
Some tools for scheduling are Apache Airflow, or Luigi.

## 12. Summary

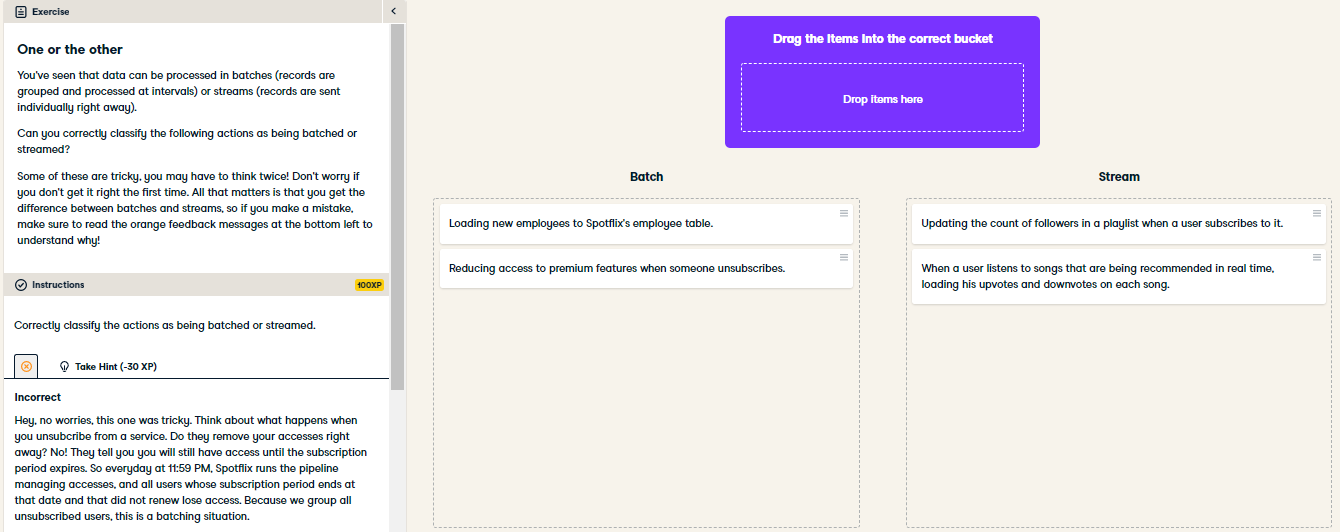
Alright! Now you know what scheduling is, the different ways to set it up, the difference between batches and streams, how scheduling is implemented at Spotflix, and a couple of tools used to schedule data engineering systems.

## 13. Let's practice!

Let's check your understanding!



Excellent! You definitely know how to schedule data processing tasks, what the different types of schedules are, and when they are appropriate. Let's go to the next exercise to see how well you understand batching and streaming!



OK! Once again, don't worry if you got some of it wrong. What matters is that hopefully, you now understand why these situations were categorized the way they were. In a nutshell, batch processing can be scheduled when resources aren't being used elsewhere, which makes it cheap. Stream processing, on the other hand is used if you need the latest data: the urgency of showing the latest data help you decide if you need it.