Writing Basic Apex Triggers

David Liu
SALESFORCE TECHNICAL ARCHITECT

@dvdkliu   sfdc99.com

WARNING:

There will be code. There will be LOTS of code.

Overview

Demo: Write a basic Apex trigger

- Line by line explanation
- See our trigger in action

The trigger loop, explained

Demo: Write a trigger!

When to use before vs. after triggers

Demo: Write a trigger!

Demo

Write a trigger that sets these lead fields

- First Name: “Hello”
- Last Name: “World”

The Trigger Loop, Explained

What Is the Trigger Loop?

```
for (Lead l : Trigger.new) {
    // We’re inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there’s fancy code here)
}
```

What Is the Trigger Loop?

```
for (Lead l : Trigger.new) {
    // We’re inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there’s fancy code here)
}
```

What Is the Trigger Loop?

```
for (Lead l : Trigger.new) {
    // We're inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there's fancy code here)
}
```

What Is the Trigger Loop?

```
for (Lead l : Trigger.new) {
    // We're inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there's fancy code here)
}
```

`Trigger.new` is the list of all records entering a trigger

What is the Trigger Loop?

```
for (Lead l : Trigger.new) {
    // We're inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there's fancy code here)
}
```

What is the Trigger Loop?

```
for (Lead l : Trigger.new) {
    // We’re inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there’s fancy code here)
}
```

What is the Trigger Loop?

```
for (Lead l : Trigger.new) {
    // We’re inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there’s fancy code here)
}
```

What is the Trigger Loop?

```
for (Lead myLead : Trigger.new) {
    // We're inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there's fancy code here)
}
```

What is the Trigger Loop?

```
for (Lead taylorSwift : Trigger.new) {
    // We're inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there's fancy code here)
}
```

What is the Trigger Loop?

```
for (Lead l : Trigger.new) {
    // We’re inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there’s fancy code here)
}
```

What is the Trigger Loop?

```
for (Lead l : Trigger.new) {
    // We're inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there's fancy code here)
}
```

What is the Trigger Loop?

```
for (Lead l : Trigger.new) {
    // We're inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there's fancy code here)
}
```

What is the Trigger Loop?

```
for (Lead l : Trigger.new) {
    // We’re inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there’s fancy code here)
}
```

What is the Trigger Loop?

```
for (Lead l : Trigger.new) {
    // We’re inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there’s fancy code here)
}
```

What is the Trigger Loop?

```
for (Lead l : Trigger.new) {
    // We're inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there's fancy code here)
}
```

What is the Trigger Loop?

```
for (Lead l : Trigger.new) {
    // We're inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there's fancy code here)
}
```

What is the Trigger Loop?

```
for (Lead l : Trigger.new) {
    // We're inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there's fancy code here)
}
```

What is the Trigger Loop?

```
for (Lead l : Trigger.new) {
    // We're inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there's fancy code here)
}
```

What is the Trigger Loop?

```
for (Lead l : Trigger.new) {
    // We’re inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there’s fancy code here)
}
```

What is the Trigger Loop?

```
for (Lead l : Trigger.new) {
    // We're inside the trigger loop!
    // Your main trigger logic will always be inside here
    // Every trigger uses a trigger loop
    // (Pretend there's fancy code here)
}
```

Why Do We Need a Trigger Loop?

Users edit in bulk

Import/delete leads

Transfer accounts

Enhanced list views

Data Loader

Workbench

PeopleImport

Dataloader.io

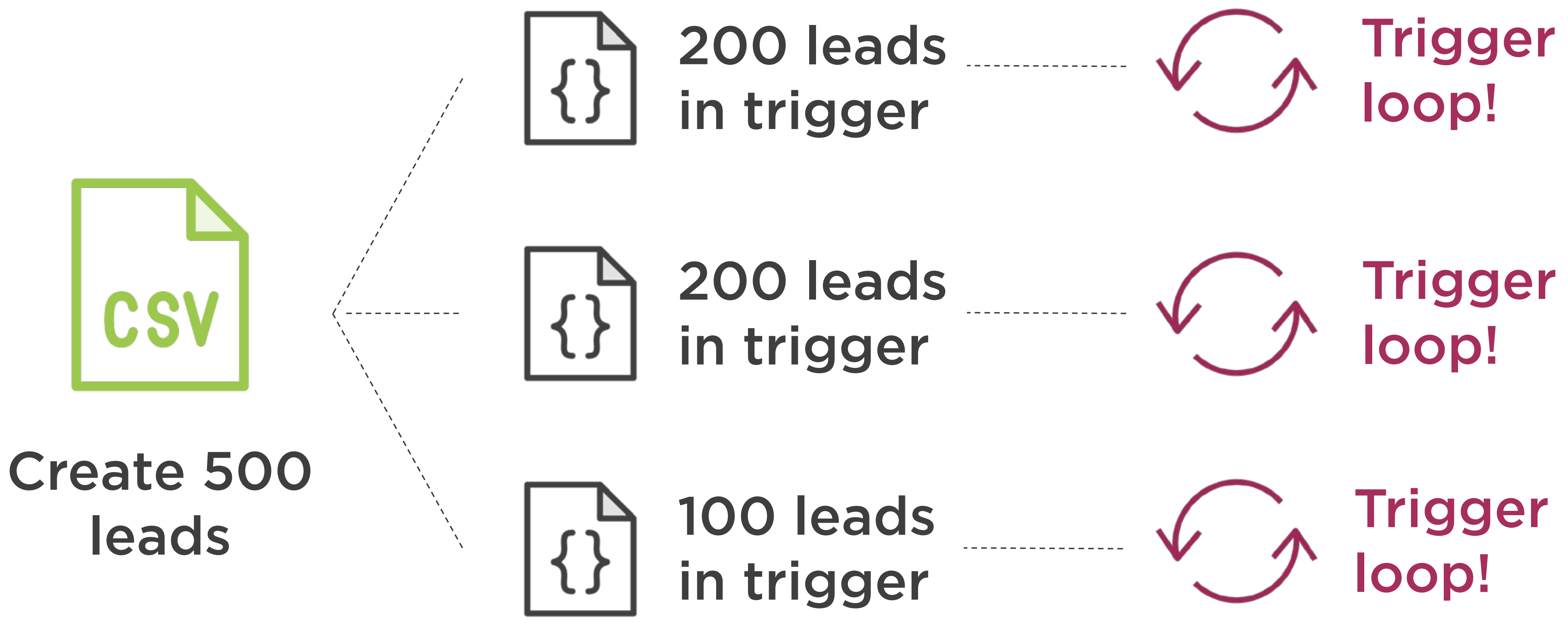
Bulk Code Updates

Trigger

Visualforce

Integrations

How the Trigger Loop Works



Demo

Write a trigger that creates a task for each new opportunity:

- Subject: “Apple Watch Promo”
- Description: “Send one ASAP!”
- Priority: “High”
- Related To: our opportunity

Comparing Our Two Triggers

```
trigger HelloWorld on Lead
(before update) {

  for (Lead l : Trigger.new) {


    l.FirstName = ‘Hello’;
    l.LastName  = ‘World’;



  }
}
```

```
trigger AppleWatch on Opportunity
(after insert) {

  for (Opportunity opp : Trigger.new) {

    Task t = new Task();

    t.Subject     = ‘Apple Watch Promo’;
    t.Description = ‘Send one ASAP!’;
    t.Priority    = ‘High’;
    t.WhatId      = opp.Id;

    insert t;
  }
}
```

When to Use “before” vs “after” Triggers

Timeline of a Database Event

PROS

- No need to explicitly save your work, the save event is coming

CONS

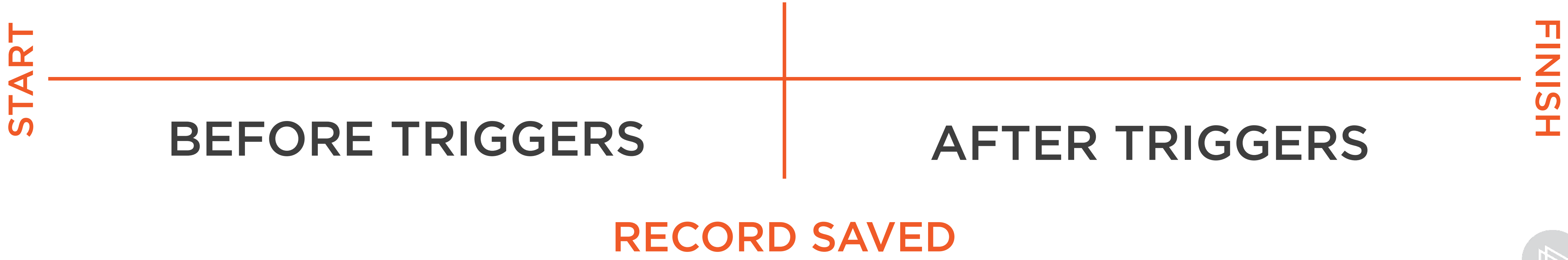
- System level fields are not available, they're not populated

PROS

- System fields now available:
  - Record ID (insert)
  - Created Date (insert)
  - Last Modified Date (update)

CONS

- Need to explicitly save your changes
- Potentially create infinite loops



When in doubt, use a "before" trigger.

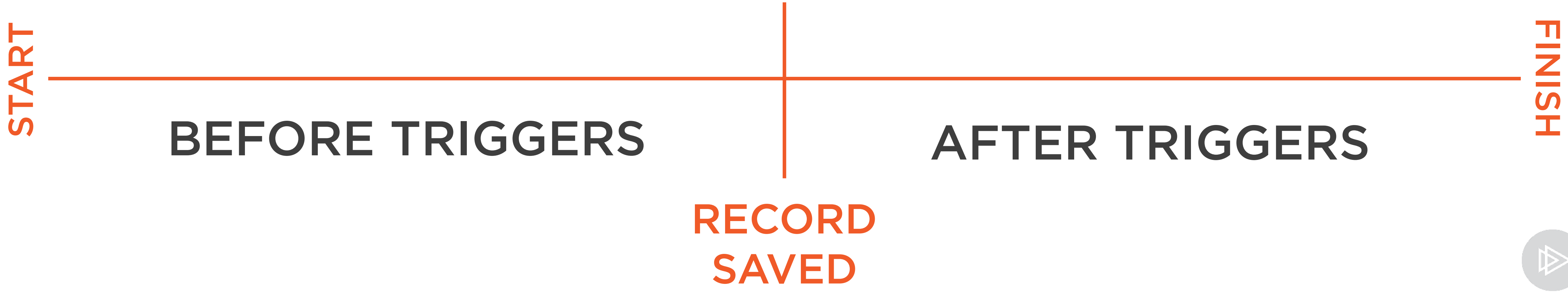
Demo

“The Infinite Looping Trigger”

Note: do not code like this!

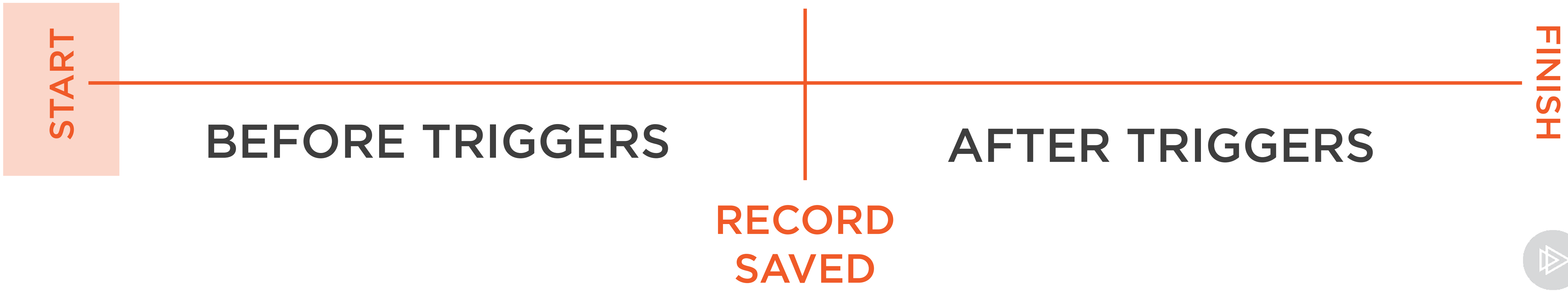
Why Would This Trigger Loop Infinitely?

```
trigger Infinity on Opportunity (after update) {
     for (Opportunity opp : Trigger.new) {
          opp.Amount = 1000;
          update opp;
     }
}
```



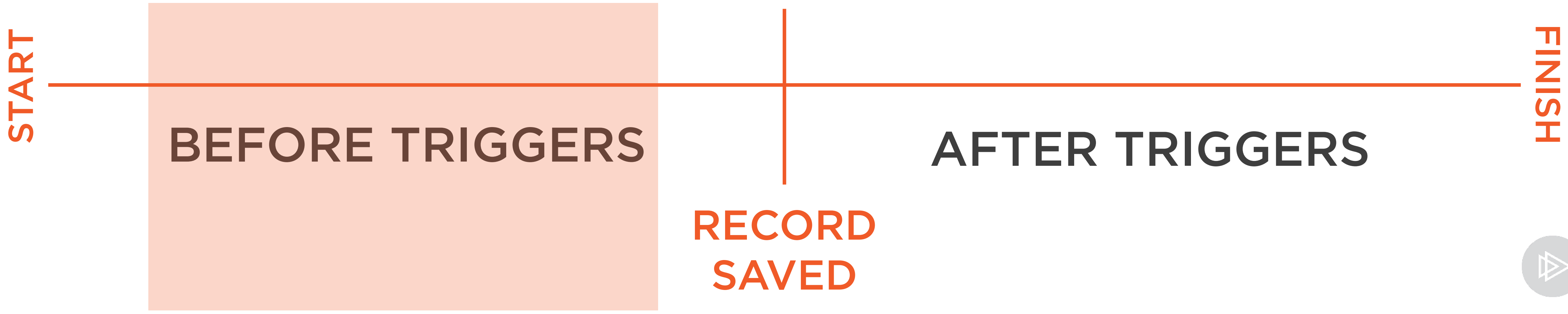
Why Would This Trigger Loop Infinitely?

```
trigger Infinity on Opportunity (after update) {
    for (Opportunity opp : Trigger.new) {
        opp.Amount = 1000;
        update opp;
    }
}
```



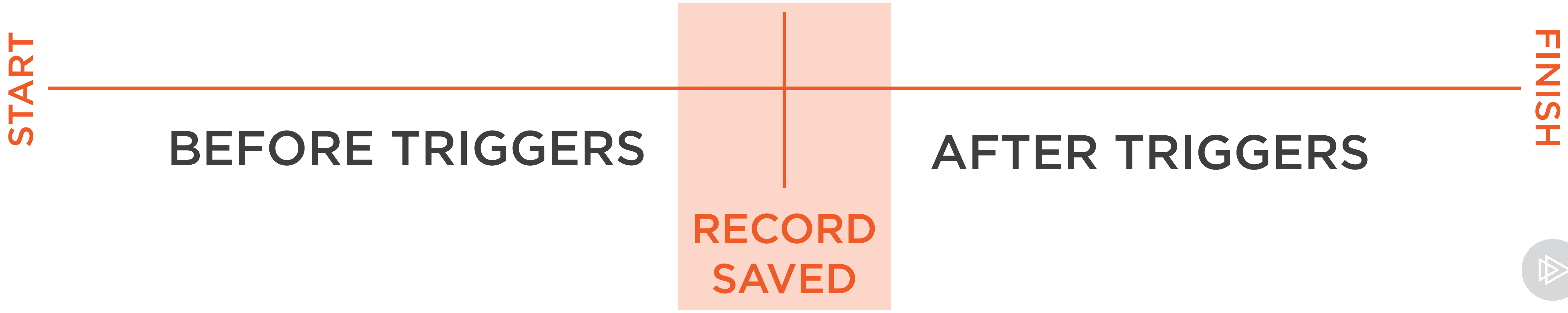
Why Would This Trigger Loop Infinitely?

```
trigger Infinity on Opportunity (after update) {
    for (Opportunity opp : Trigger.new) {
        opp.Amount = 1000;
        update opp;
    }
}
```



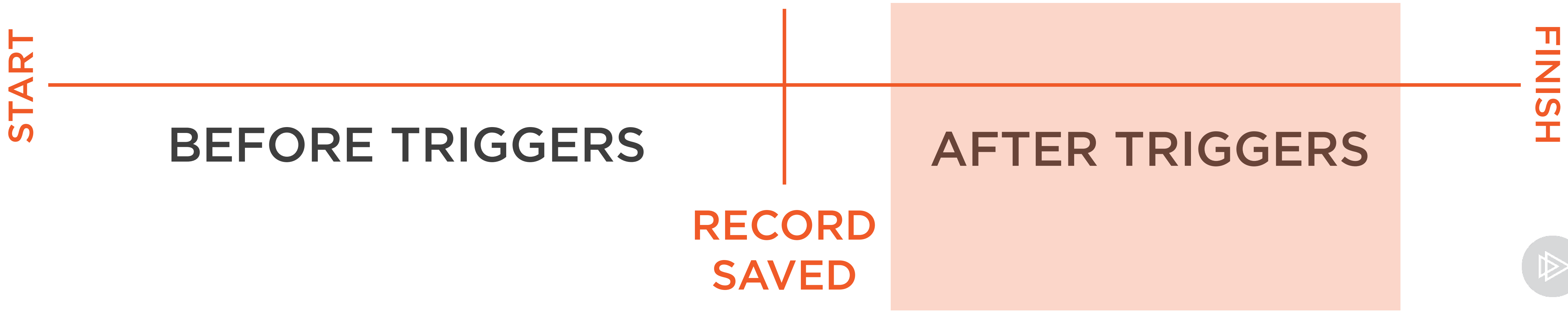
Why Would This Trigger Loop Infinitely?

```
trigger Infinity on Opportunity (after update) {
    for (Opportunity opp : Trigger.new) {
        opp.Amount = 1000;
        update opp;
    }
}
```



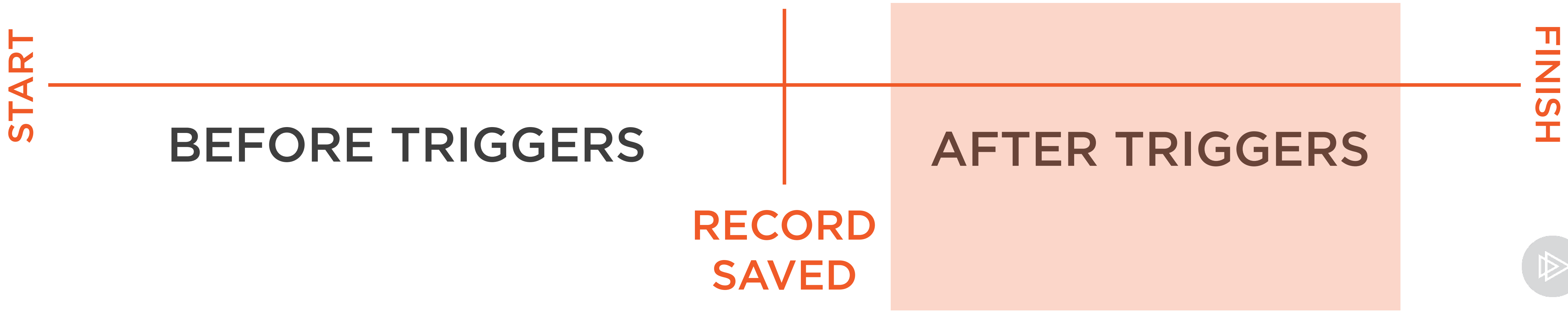
Why Would This Trigger Loop Infinitely?

```
trigger Infinity on Opportunity (after update) {
    for (Opportunity opp : Trigger.new) {
        opp.Amount = 1000;
        update opp;
    }
}
```



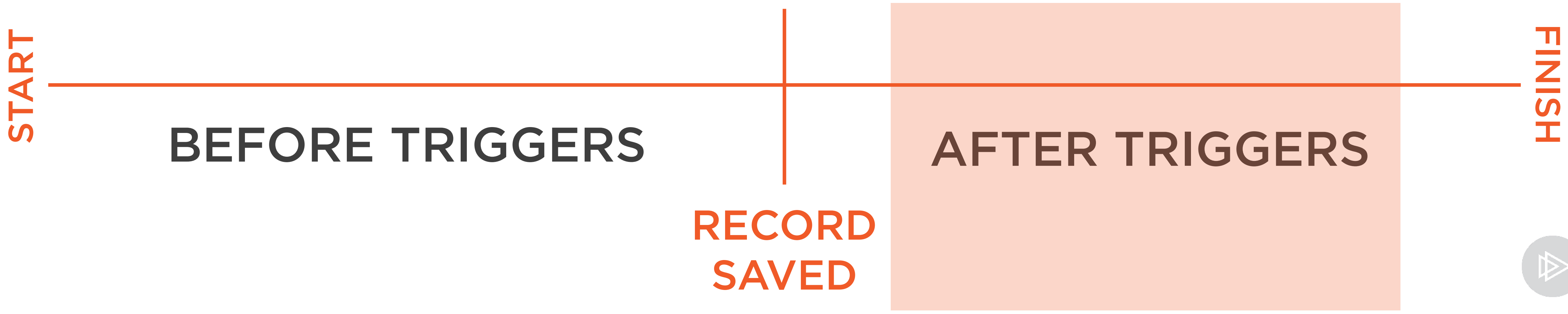
Why Would This Trigger Loop Infinitely?

```
trigger Infinity on Opportunity (after update) {
     for (Opportunity opp : Trigger.new) {
          opp.Amount = 1000;
          update opp;
     }
}
```



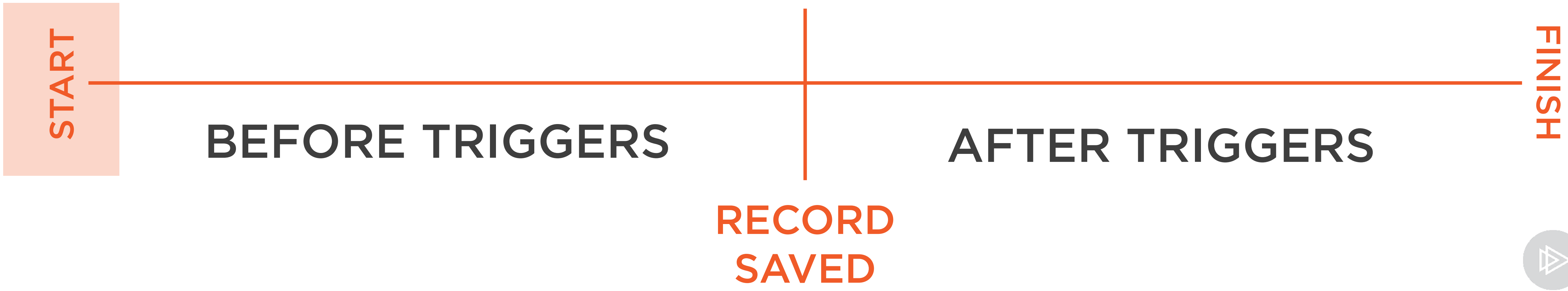
Why Would This Trigger Loop Infinitely?

```
trigger Infinity on Opportunity (after update) {
    for (Opportunity opp : Trigger.new) {
        opp.Amount = 1000;
        update opp;
    }
}
```



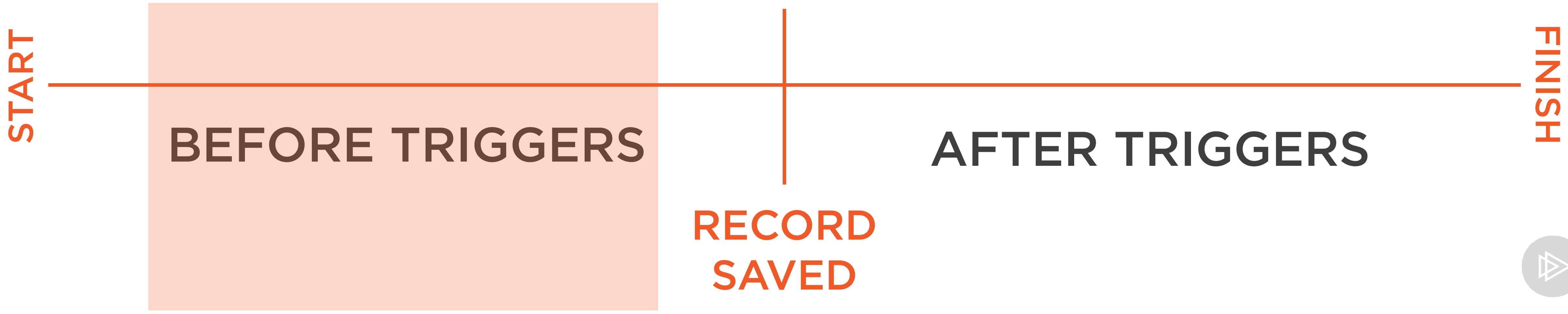
Why Would This Trigger Loop Infinitely?

```
trigger Infinity on Opportunity (after update) {
    for (Opportunity opp : Trigger.new) {
        opp.Amount = 1000;
        update opp;
    }
}
```



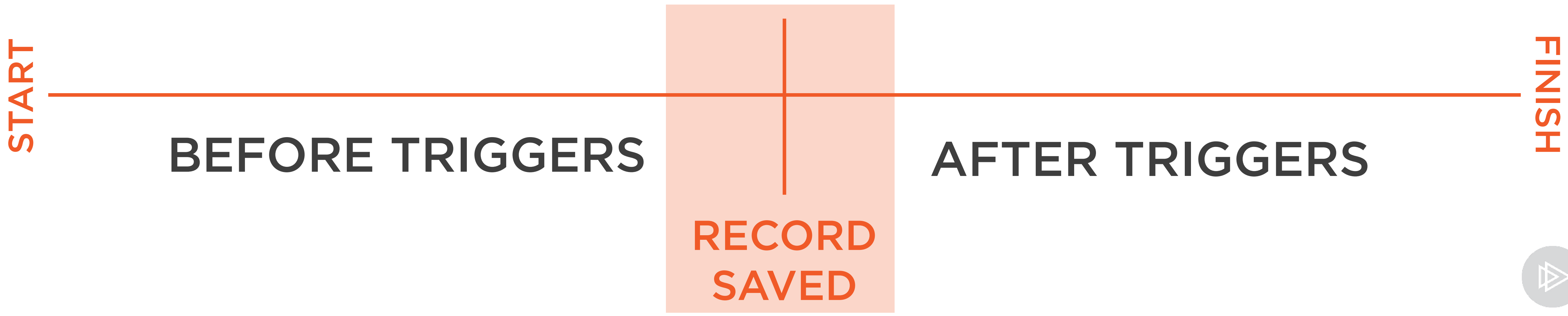
Why Would This Trigger Loop Infinitely?

```
trigger Infinity on Opportunity (after update) {
    for (Opportunity opp : Trigger.new) {
        opp.Amount = 1000;
        update opp;
    }
}
```



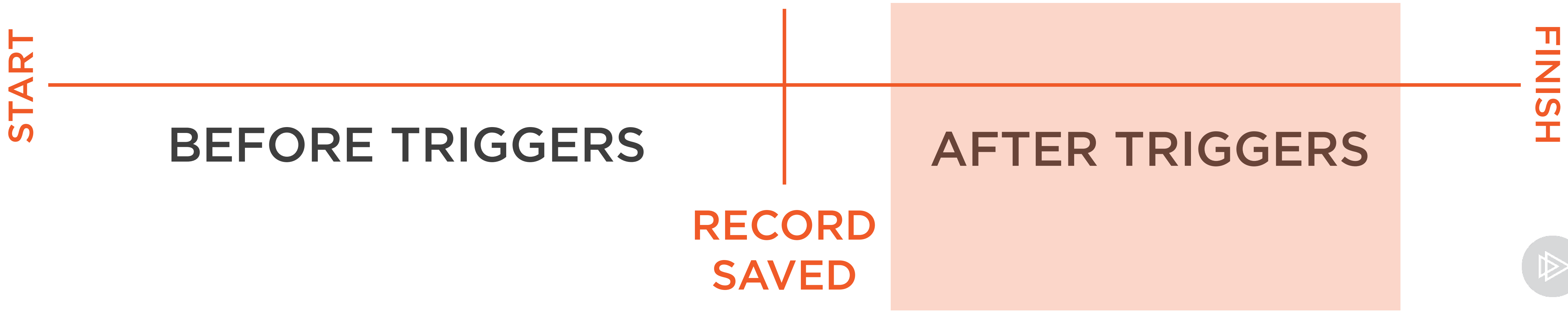
Why Would This Trigger Loop Infinitely?

```
trigger Infinity on Opportunity (after update) {
    for (Opportunity opp : Trigger.new) {
        opp.Amount = 1000;
        update opp;
    }
}
```



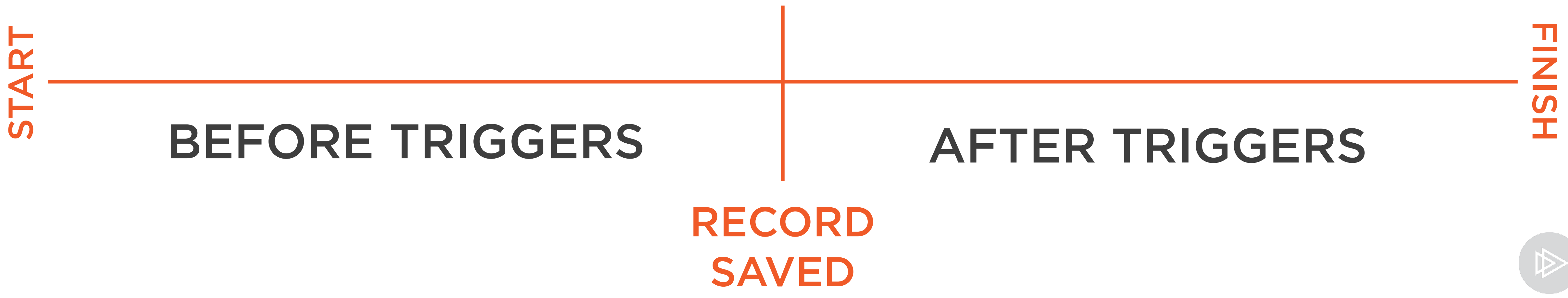
Why Would This Trigger Loop Infinitely?

```
trigger Infinity on Opportunity (after update) {
    for (Opportunity opp : Trigger.new) {
        opp.Amount = 1000;
        update opp;
    }
}
```



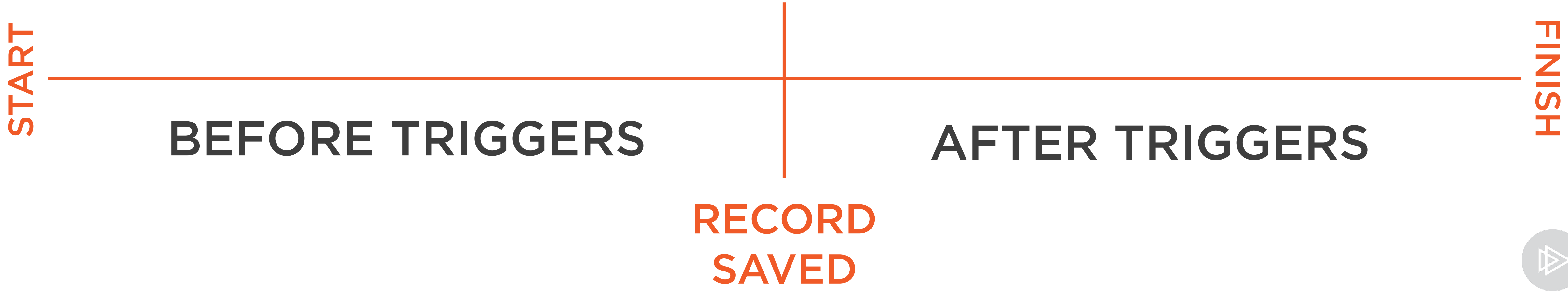
How to Fix the Infinite Trigger

```
trigger Infinity on Opportunity (after update) {
        for (Opportunity opp : Trigger.new) {
                opp.Amount = 1000;
                update opp;
        }
}
```



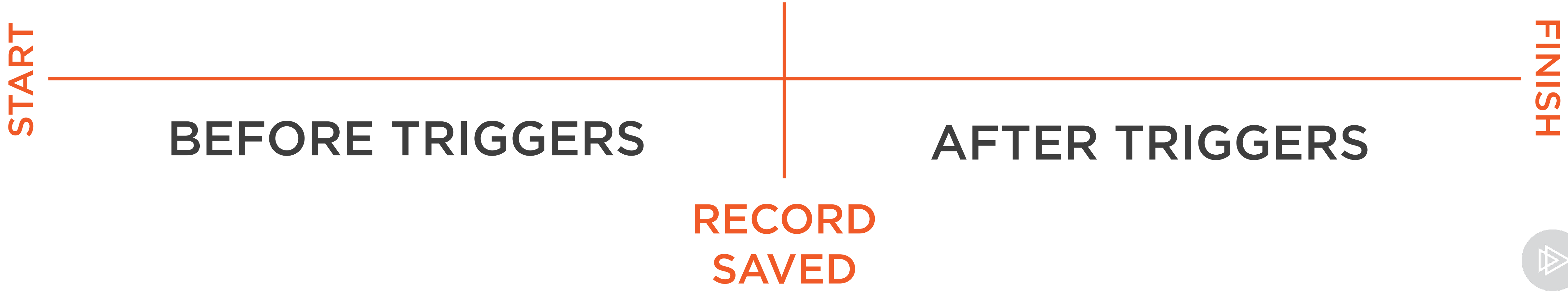
How to Fix the Infinite Trigger

```
trigger Infinity on Opportunity (before update) {
    for (Opportunity opp : Trigger.new) {
        opp.Amount = 1000;
        update opp;
    }
}
```



How to Fix the Infinite Trigger

```
trigger Infinity on Opportunity (before update) {
    for (Opportunity opp : Trigger.new) {
        opp.Amount = 1000;
        ~~update opp;~~
    }
}
```



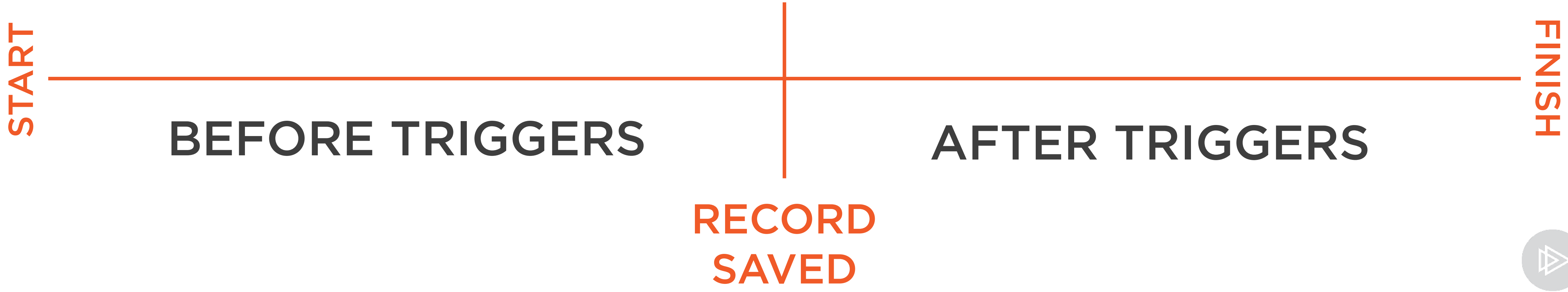
Demo

“The Non-Existent ID”

Note: do not code like this!

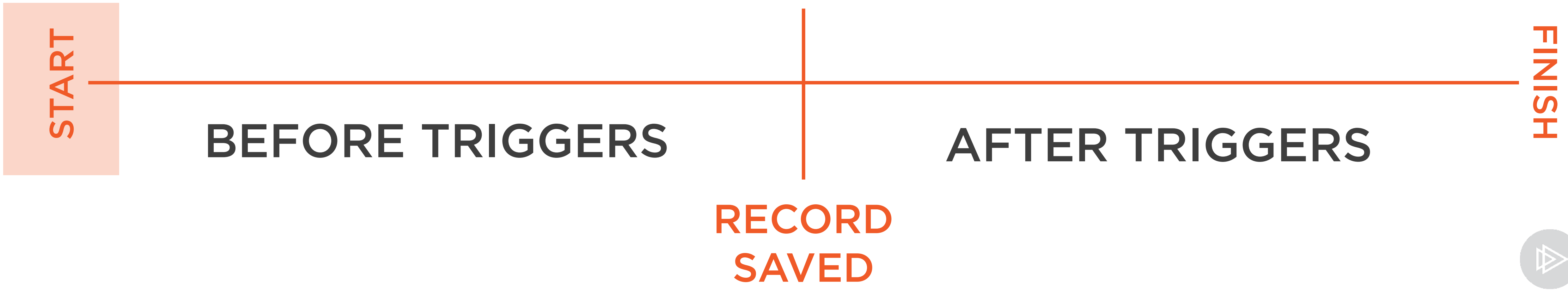
Why Isn’t the Record ID Available?

```
trigger NonExistentId on Case (before insert) {
     for (Case myCase : Trigger.new) {
          CaseComment cc = new CaseComment();
          cc.CommentBody = ‘Case received by Agent’;
          cc.ParentId    = myCase.Id;
          insert cc;
     }
}
```



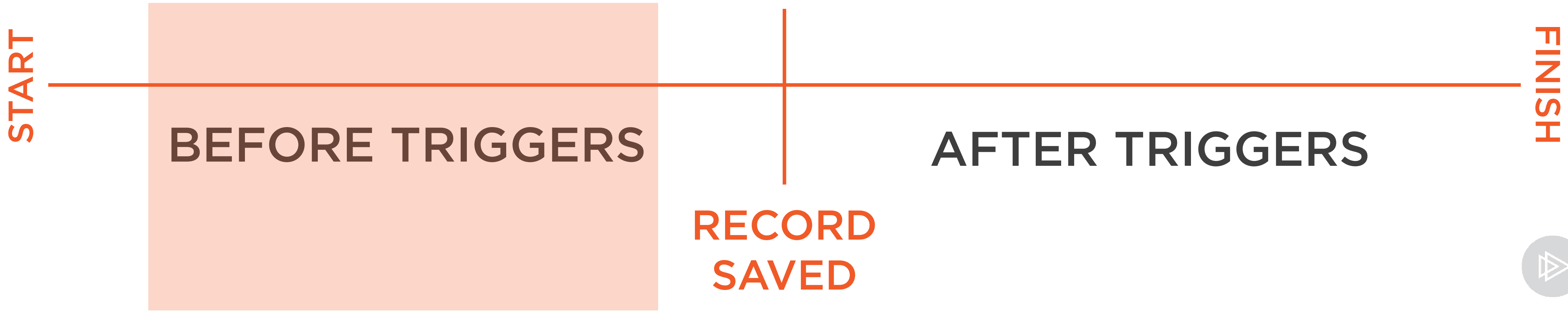
Why Isn’t the Record ID Available?

```
trigger NonExistentId on Case (before insert) {
     for (Case myCase : Trigger.new) {
          CaseComment cc = new CaseComment();
          cc.CommentBody = ‘Case received by Agent’;
          cc.ParentId    = myCase.Id;
          insert cc;
     }
}
```



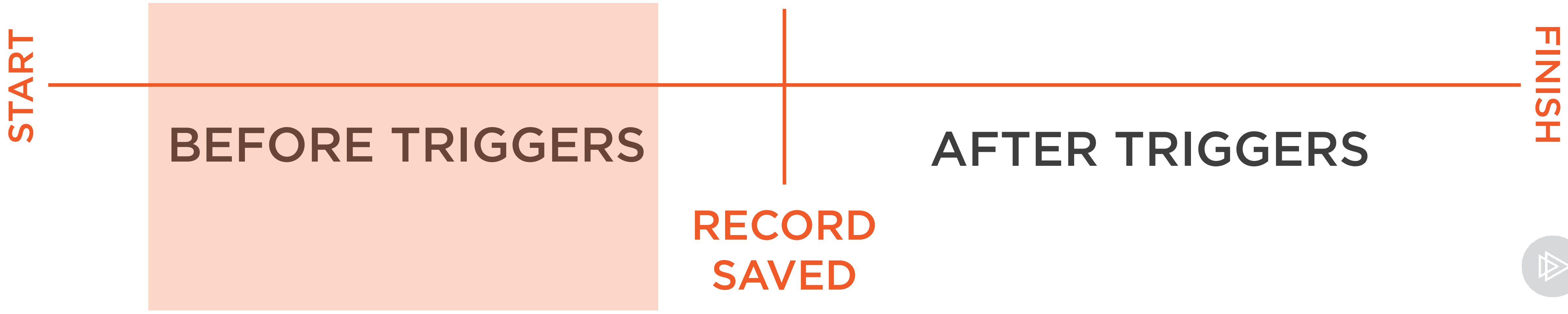
Why Isn't the Record ID Available?

```
trigger NonExistentId on Case (before insert) {
     for (Case myCase : Trigger.new) {
          CaseComment cc = new CaseComment();
          cc.CommentBody = ‘Case received by Agent’;
          cc.ParentId    = myCase.Id;
          insert cc;
     }
}
```



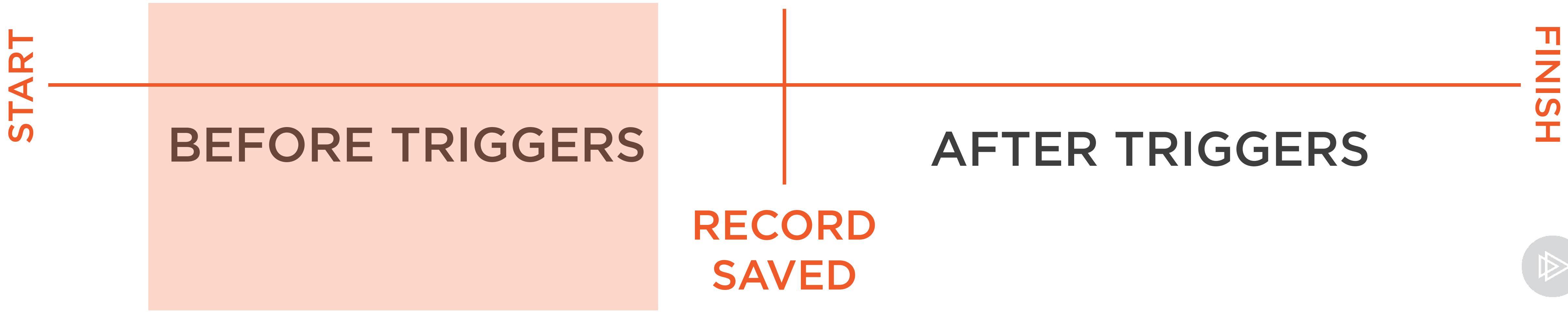
Why Isn’t the Record ID Available?

```
trigger NonExistentId on Case (before insert) {
    for (Case myCase : Trigger.new) {
        CaseComment cc = new CaseComment();
        cc.CommentBody = ‘Case received by Agent’;
        cc.ParentId    = myCase.Id;
        insert cc;
    }
}
```



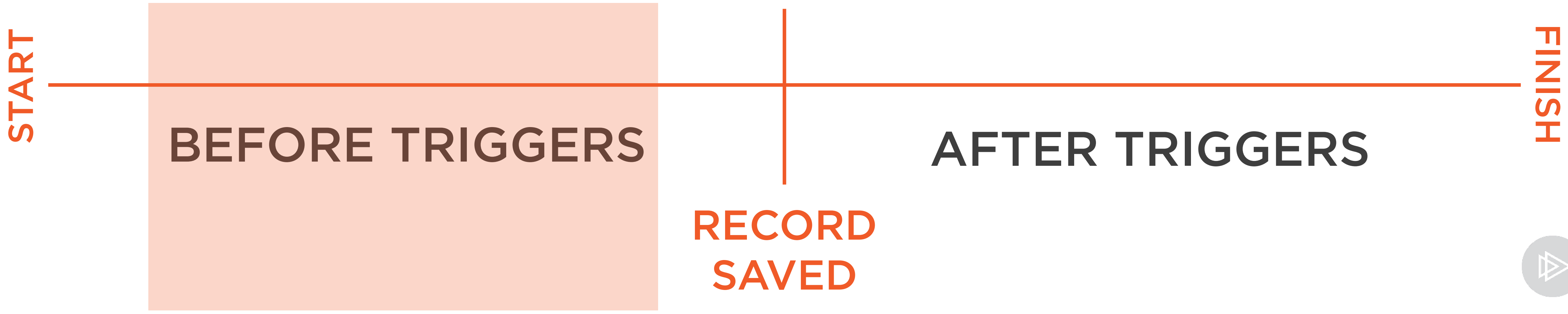
Why Isn’t the Record ID Available?

```
trigger NonExistentId on Case (before insert) {
    for (Case myCase : Trigger.new) {
        CaseComment cc = new CaseComment();
        cc.CommentBody = ‘Case received by Agent’;
        cc.ParentId    = myCase.Id;
        insert cc;
    }
}
```



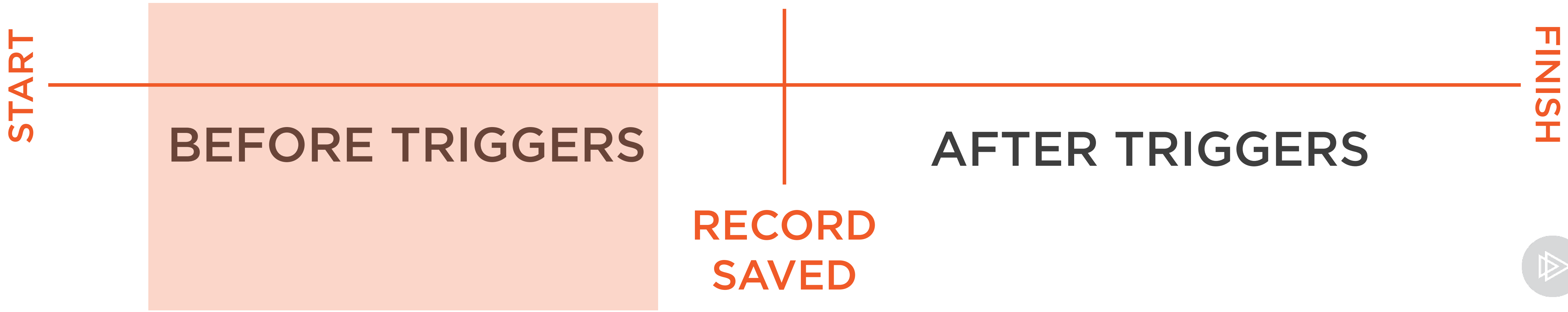
Why Isn’t the Record ID Available?

```
trigger NonExistentId on Case (before insert) {
    for (Case myCase : Trigger.new) {
        CaseComment cc = new CaseComment();
        cc.CommentBody = ‘Case received by Agent’;
        cc.ParentId    = myCase.Id;
        insert cc;
    }
}
```



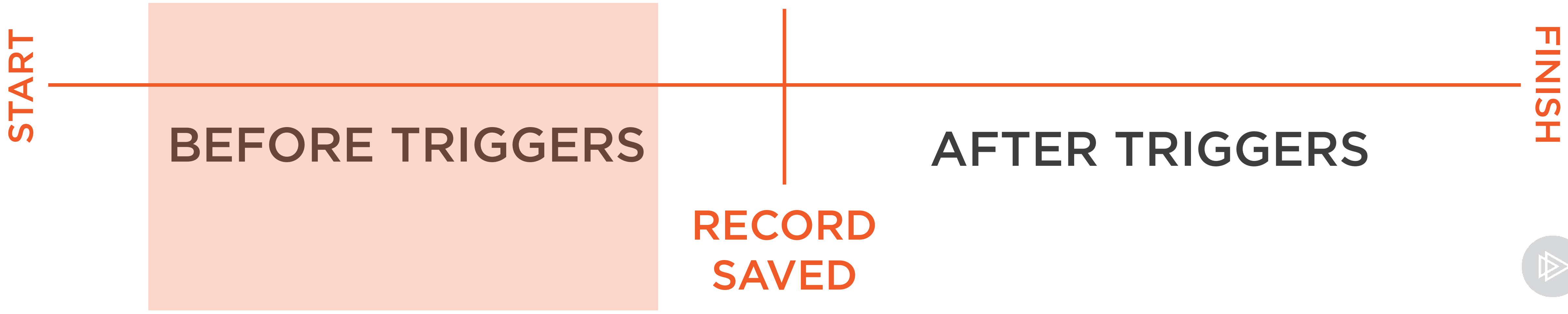
Why Isn’t the Record ID Available?

```
trigger NonExistentId on Case (before insert) {
    for (Case myCase : Trigger.new) {
        CaseComment cc = new CaseComment();
        cc.CommentBody = ‘Case received by Agent’;
        cc.ParentId    = myCase.Id;
        insert cc;
    }
}
```



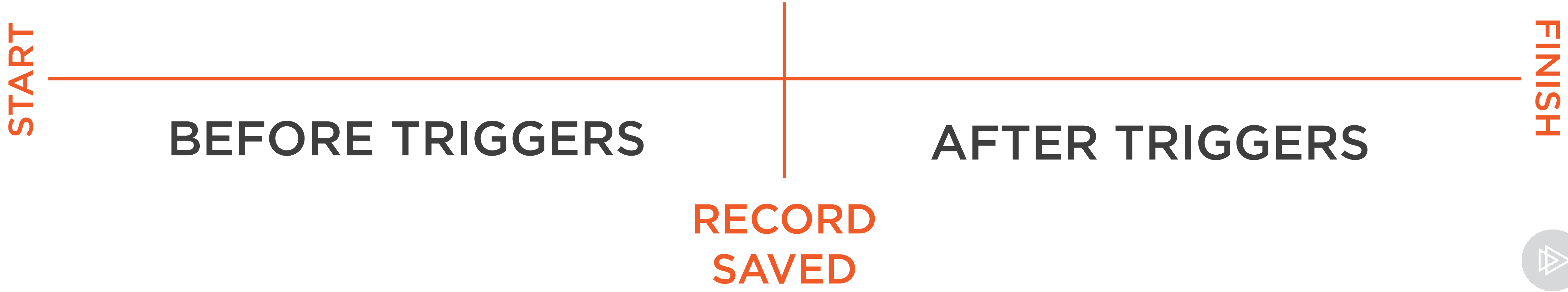
Why Isn’t the Record ID Available?

```
trigger NonExistentId on Case (before insert) {
     for (Case myCase : Trigger.new) {
          CaseComment cc = new CaseComment();
          cc.CommentBody = ‘Case received by Agent’;
          cc.ParentId    = myCase.Id;
          insert cc;
     }
}
```



How to Fix The Non-Existent ID

```
trigger NonExistentId on Case (after insert) {
    for (Case myCase : Trigger.new) {
        CaseComment cc = new CaseComment();
        cc.CommentBody = ‘Case received by Agent’;
        cc.ParentId    = myCase.Id;
        insert cc;
    }
}
```



The General Trigger Pattern

```
trigger <<Name>> on <<Object>> (<<before/after>> <<event>>) {
   for (<<Object>> <<variable>> : Trigger.new) {
      // Create an object (optional)
      // Update record
      // Explicit save (optional)
   }
}
```

Demo

Write a trigger that does the following when an Account is created:

- Create a new case
- Assign the owner to your new intern
- Subject: Dedupe this account
- Associate with the account

Summary

How to write a basic trigger

- The general trigger pattern
- No need to reinvent the wheel!

The trigger loop, explained

- You'll always use the trigger loop
- Multiple records can enter at once

"Before" vs "after" triggers

- When in doubt, use "before"
- System fields are available in "after"