

# Introduction

- We already covered how to update *one* document at a time
- Let's now update *multiple* documents within a single query
  - Similar to an `UPDATE WHERE` query in a RDBMS
- The query uses three concepts that we have just covered
  - Primary terms
  - Sequence numbers
  - Optimistic concurrency control

# How the snapshot is used

- Prevents overwriting changes made *after* the snapshot was taken
  - The query may take a while to finish if updating many documents
- Each document's primary term and sequence number is used
  - A document is only updated if the values match the ones from the snapshot
  - As you know, this is called *optimistic concurrency control* 😊
- # of version conflicts is returned within the `version_conflicts` key

# Lecture summary (1/2)

- The query creates a snapshot to do optimistic concurrency control
- Search queries and bulk requests are sent to replication groups sequentially
  - Elasticsearch retries these queries up to ten times
  - If the queries *still* fail, the whole query is aborted
    - Any changes already made to documents, are *not* rolled back
- The API returns information about failures

## Lecture summary (2/2)

- If a document has been modified since taking the snapshot, the query is aborted
  - This is checked with the document's primary term and sequence number
- To count version conflicts instead of aborting the query, the `conflicts` option can be set to `proceed`