In this system, the **visibility timeout** and **Dead-Letter Queue (DLQ)** were key mechanisms to ensure reliable, consistent, and fault-tolerant message processing.

Visibility Timeout:

The visibility timeout prevents multiple consumers from processing the same message simultaneously. When the Lambda function retrieves a message from the SQS queue, that message becomes hidden from other consumers for the visibility timeout period. If the Lambda function completes successfully and deletes the message, no other consumer will see it. However, if the function fails or doesn't delete the message before the timeout expires, the message becomes visible again for retry.

In this project, visibility timeout ensured that:

- Each order message was processed only once at a time.
- Messages that weren't processed successfully were retried after the timeout.

This protected against duplicate inserts into DynamoDB and avoided race conditions.

Dead-Letter Queue (DLQ):

The DLQ captured messages that couldn't be processed successfully after a set number of attempts (3 in this setup). Once a message failed processing 3 times, it was moved to the DLQ for later analysis rather than staying in the main queue.

The DLQ was valuable because:

- It prevented continuously failing messages from blocking the main queue.
- It provided a place to isolate and investigate problematic messages (e.g., invalid data).
- It allowed monitoring of processing errors without data loss.