**Issue Summary**

**Duration:**  
The outage occurred on August 10, 2024, starting at 10:45 AM UTC and ending at 1:15 PM UTC, lasting a total of 2 hours and 30 minutes.

**Impact:**  
During the outage, our main e-commerce platform experienced severe slowdowns, leading to unresponsive pages and failed transactions. Approximately 75% of our users were affected, resulting in a 50% decrease in successful purchases during the outage period.

**Root Cause:**  
The root cause was traced back to a misconfigured database query in our payment processing service. This query caused a significant increase in CPU utilization on the database server, leading to a bottleneck that affected the entire application stack.

**Timeline**

* **10:45 AM UTC:** Issue detected by a spike in error rates and a drop in successful transaction counts as reported by monitoring tools.
* **10:50 AM UTC:** On-call engineer received alerts and began investigating the application logs and database performance metrics.
* **11:00 AM UTC:** Initial investigation focused on a suspected network issue due to latency spikes, but no anomalies were found.
* **11:20 AM UTC:** The issue was escalated to the database team after high CPU utilization on the database server was identified.
* **11:30 AM UTC:** The database team began analyzing recent query performance and identified a specific query with unusually long execution times.
* **12:00 PM UTC:** The query was identified as originating from the payment processing service. The service was temporarily disabled to prevent further strain on the database.
* **12:30 PM UTC:** A hotfix was developed and deployed to optimize the problematic query.
* **1:00 PM UTC:** The payment processing service was re-enabled, and monitoring indicated a return to normal performance levels.
* **1:15 PM UTC:** The incident was officially resolved, and a postmortem meeting was scheduled.

**Root Cause and Resolution**

The issue was caused by an inefficient SQL query introduced in a recent update to the payment processing service. The query, intended to aggregate transaction data, lacked proper indexing, leading to full table scans on a large, frequently accessed table. This caused a significant increase in CPU load on the database server, leading to slow response times across the application.

The resolution involved temporarily disabling the payment processing service to alleviate the load on the database while a hotfix was developed. The hotfix included adding the necessary indexes to the table and optimizing the query to reduce its complexity. After deploying the hotfix, the payment processing service was re-enabled, and the system returned to normal operation.

**Corrective and Preventative Measures**

**Improvements/Fixes:**

* Implemented a thorough review process for database queries before deployment.
* Added monitoring on database query performance to detect inefficiencies early.
* Enhanced logging to provide better insights into query performance and application behavior during high-load periods.

**Tasks:**

* Patch the payment processing service with the optimized query.
* Add appropriate indexes to the affected database table.
* Update the deployment process to include query performance testing.
* Implement automated alerts for high CPU utilization on database servers.
* Conduct a training session for developers on SQL query optimization and database performance best practices.

By addressing these measures, we aim to prevent similar issues from occurring in the future and ensure a more robust and responsive system for our users.