Module	**Business Database Management**
Assessment	Continuous Assessment Two (CA2)
Title	**Comparative Analysis of Database Models for E-commerce & GDPR Risk Assessment**
Group Members	Pushmita Kharat (20070446)
	Subasri Nandakumar (20079001)
	Sharkesh Raja (20084066)
	Sreelakshmi Sureshkumar (20074696)
	Eren Kurtulus (20085675)
Submission Date	12-12-2025

2. Introduction

The pace of change in e-commerce is relentless, and companies constantly search for smarter ways to analyze customer behavior. The typical online shopper generates numerous interactions—viewing, adding to cart, saving for later, and purchasing—and tracking all these actions and making sense of them is a significant challenge for businesses. While relational databases have traditionally stored this information, their structure can become tangled when trying to connect the dots in a customer's journey. Graph databases, such as Neo4j, offer a fresh perspective by putting the spotlight on relationships rather than spreading information across a maze of tables.

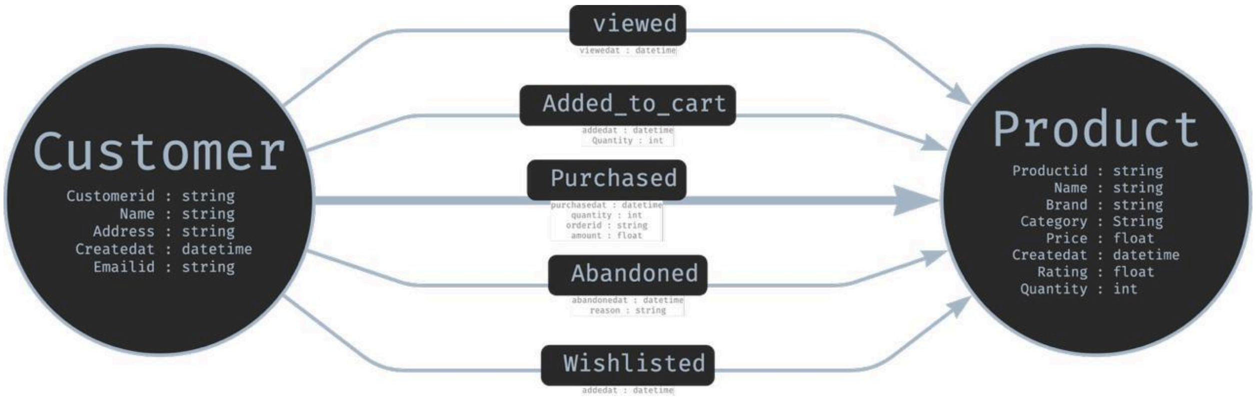
This report outlines the design of a specialized database solution using **Neo4j**, a graph database, to model and analyze complex customer journeys for an e-commerce platform. It provides a comparative analysis between the relational and graph database approaches for this specific use case, demonstrating how graph databases excel at generating actionable insights through relationship-centric modeling. Furthermore, it assesses the data governance and security implications, focusing on compliance with the **General Data Protection Regulation (GDPR)**.

3. Database Schema: Conceptual Graph Model

The model is built on two main entities: **Customer** and **Product**.

- **Customer** nodes include essentials like their ID, name, address, signup date, and email.
- **Product** nodes have their own set of details—ID, name, brand, category, price, when they were added, rating, and stock levels.

The relationships between them capture the entire shopping path. Every time a customer interacts with a product, a new connection is formed.



The relationship lines hold valuable details like the time of the action, the quantity involved, or even the reason a cart was left behind.

4. Part A: Comparative Analysis of Database Models

Relational databases have long been the backbone for storing this information. They're dependable and handle transactions well, but when it comes to connecting the dots in a customer's journey, things can get tangled fast.

In the relational model, there are separate tables for customers, products, views, cart items, orders, order lines, abandoned carts, and wishlists. Each action is just a row in a table, tied to others by foreign keys. While this system is solid for basic reporting, it quickly becomes a headache when someone wants to answer relationship-based questions like, "What do people usually buy after purchasing a certain item?" or "What do customers buy after abandoning their cart?". Dealing with a tangle of joins, self-joins, and complicated queries, which leads to a "JOIN explosion" as data grows, is required.

The graph database approach solves this by making every interaction a direct, traceable relationship. The structure is refreshingly straightforward. The real advantage of using a graph database comes into focus when looking at the kinds of questions businesses actually care about.

For example, to know what's most often bought after a specific product, one can just follow the `PURCHASED` links from that product to see what else those customers picked up next. In SQL, the order lines table must be joined to itself, sorted by purchase time, and the query may slow down. Looking for recommendations based on what similar customers buy is easy with a graph, by spotting shoppers who bought the same things, then seeing what else they chose that the original customer hasn't tried yet. This collaborative filtering is a natural fit for graphs.

Abandoned carts are another area where graphs shine. Neo4j makes it possible to follow the `ABANDONED` link from a customer to a product, then see what they ended up buying instead, and when. Spotting product pairs that are frequently bought together is also much simpler in a graph. Just look for customers with multiple `PURCHASED` links in the same order, group the pairs, and see which ones are most common.

Scalability is another big plus for graph databases. The time it takes to follow a path depends on how many steps there are, not how big the whole database is. Relational databases, on the other hand, slow down as their tables get bigger and the joins pile up. That said, relational databases are still great for transactions, keeping data consistent, and generating standard reports. Many companies use a mix: SQL for the core transactional data, and a graph database for analysing behaviour and making recommendations. In the end, the kinds of insights that matter most in e-commerce are much easier to get from a graph database like Neo4j.

5. Part B: Data Governance and GDPR Risk Analysis

This section integrates the content from DBMS Ca2.docx (Member 5's work).

The General Data Protection Regulation (GDPR) has revolutionized how data is governed, particularly for entities like e-commerce businesses dealing with large amounts of personal data. The regulation mandates robust data protection measures, transparency, and accountability, which are critical for any database design.

Key GDPR Principles and Database Implications

1. **Lawfulness, Fairness, and Transparency:**
   - The collection and use of personal data must be clear and justified.
   - Risk: The detailed customer journey mapping in Neo4j (e.g., VIEWED, ABANDONED relationships) could be deemed excessive if not clearly consented to, violating the principle of data minimization.
2. **Purpose Limitation and Data Minimisation:**
   - Data should only be processed for specified, explicit, and legitimate purposes.
   - Risk: In the graph model, the properties attached to relationships, such as the reason for abandonment or detailed time stamps, must be strictly relevant. Holding onto historic viewing data for too long without a clear purpose can violate this principle.
3. **Accuracy and Storage Limitation:**
   - Personal data must be accurate and kept up to date. Data should also not be kept longer than necessary.
   - Risk: The graph model, which permanently links customers to their entire journey, makes anonymization and deletion difficult. Automated data lifecycle management policies to pseudonymise or delete old relationship data are necessary.
4. **Integrity and Confidentiality (Security):**
   - Personal data must be secured against unlawful processing, accidental loss, destruction, or damage.

    - Risk: Graph databases storing sensitive customer details (Emailid, Address) require strong access controls and encryption. The interconnected nature of the graph means a breach in one area could expose a vast network of personal activity.

Data Subject Rights and Graph Databases

The GDPR grants data subjects specific rights that pose implementation challenges for complex graph structures:

- **Right to Erasure (Right to be Forgotten):** When a customer requests deletion, all their nodes and all related relationship data must be permanently removed.
- **Right to Data Portability:** Providing a customer with a copy of all the personal data they have provided requires extracting all node and relationship properties linked to that customer ID from the graph.

Risk Mitigation Strategies

- **Pseudonymisation:** Instead of storing the real Customer ID in the graph, a unique, reversible token should be used, with the key stored separately under high security.
- **Access Control:** Implementing role-based access control (RBAC) within Neo4j to restrict which users can view sensitive properties.
- **Data Minimization in Design:** Only recording necessary properties on relationships.

In conclusion, while the Neo4j graph database offers superior analytical power for e-commerce, its complex, interconnected structure amplifies GDPR risks related to data minimization, storage limitation, and the implementation of data subject rights. A proactive governance strategy incorporating pseudonymisation and rigorous access controls is essential for compliance.

---

6. Query Summary

The group developed six Cypher queries to answer crucial business questions, demonstrating the strength of the graph model in analyzing relationships.

1. **Next Purchase Prediction (Sequential Purchase Analysis):**
   - **Goal:** To find which products are most frequently bought after a specific target item.
   - **Logic:** The query starts by finding customers who purchased a target product (e.g., `P1`). It then follows the next `PURCHASED` relationship from those same customers to any other product, excluding the target itself, and aggregates the counts. This reveals common sequential shopping patterns for merchandising strategies.
2. **Collaborative Filtering (Similar Customer Recommendations):**
   - **Goal:** To generate product recommendations for a specific customer based on the purchases of similar users.
   - **Logic:** The query first identifies customers (`other`) who share purchase history with a target customer (e.g., `C1`). It then isolates products that these similar customers purchased that the target customer has *not* purchased, ranking them by frequency. This is a classic relationship-based recommendation method.
3. **Abandoned Cart Analysis (Purchase Substitution):**
   - **Goal:** To determine what alternative product a customer purchased instead of an item they abandoned.
   - **Logic:** The query isolates customers who have an `ABANDONED` relationship linked to a specific product (e.g., `P3`). It then looks for a later `PURCHASED` relationship from that customer to any other product, providing a time-stamped view of the substitution behavior to understand why the cart was abandoned.
4. **Product Affinity (Frequently Bought Together):**
   - **Goal:** To determine which two distinct products are most commonly bought in the same order/transaction.

   - **Logic:** The query matches two separate `PURCHASED` relationships (`p1`, `p2`) from the same customer that share the same order ID (`p1.orderId = p2.orderId`). It then counts and ranks these unique product pairs, identifying strong affinities for bundle creation or shelf placement optimization.

5. **High-Consideration Items (Heavy View-to-Cart Conversion):**
   - **Goal:** To identify products that require a high number of views or research before a commitment is made (adding to cart).
   - **Logic:** The query first counts the total number of `VIEWED` relationships a customer has with a product. It then filters for cases where the same customer later `ADDED_TO_CART` that product, but only if the view count exceeded a certain threshold (e.g., > 3).
6. **Impulse Purchases (Short View-to-Purchase Time):**
   - **Goal:** To identify products purchased quickly, suggesting an impulse decision or high demand.
   - **Logic:** The query calculates the time difference between the timestamp of the last `VIEWED` event and the timestamp of the final `PURCHASED` event for the same customer-product pair. It filters for transactions where this duration is extremely short (e.g., less than 10 minutes), identifying rapid, low-consideration transactions.

---

7. Conclusion

The analysis demonstrates that for the specific requirements of e-commerce behavioral tracking, a graph database like Neo4j offers significant performance and conceptual advantages over traditional relational models. By modeling every interaction as a direct relationship, complex path analysis becomes intuitive and scalable. However, this powerful interconnectedness introduces heightened governance responsibilities, particularly concerning the General Data Protection Regulation (GDPR). Successful deployment requires not only a robust graph design but also stringent security, access control, and pseudonymisation strategies to protect sensitive customer data and ensure compliance with data subject rights. A hybrid approach utilizing both SQL for transactional integrity and Neo4j for analytical insight offers the best path forward.

---

8. References

European Commission (2018) *General Data Protection Regulation (GDPR).*

Voigt, P. and von dem Bussche, A. (2017) *The EU General Data Protection Regulation (GDPR): A Practical Guide.* Cham: Springer.

Robinson, I., Webber, J. and Eifrem, E. (2015) *Graph Databases.* 2nd edn. Sebastopol: O'Reilly Media.

Information Commissioner's Office (ICO) (2021) *Guide to the General Data Protection Regulation (GDPR).*
