MANCHESTER
1824
The University of Manchester
Alliance Manchester Business School

DATA70141 Assignment 2 (group): Amazone

Your group's task is to design, implement and demonstrate a NoSQL database for the Amazone online shopping website that operates only in the UK. Each group will submit presentation slides co-design by the group members, a recorded video presentation that is between 15 minutes to 20 minutes long (each member of the group must present a minimum of one slide), and a zipped copy of their final database and queries. Each group member will also submit a personal reflective essay. If you find any ambiguities in the specification of this assignment, you should decide what to do and clearly document and justify your decisions in your group slides and presentation.

Case Background

Amazone has decided to expand its operations by delivering fresh groceries. Amazone partnered with Morrizon, a UK grocery retailer, to offer same-day and instant grocery delivery service. Amazone has decided to experiment with same-day and instant grocery delivery using Manchester. Six Morrizon grocery stores have been chosen for instant pick-up and delivery by delivery drivers who are Amazon partners.

The Amazone team has contacted your team to develop a new online platform that accommodates both the existing and new business models. Your team has received the following information to help with the data modelling and implementation service.

ER Diagram

Below is an ER diagram of the old database schema. However, your team has been informed that this schema may be slightly different from the final schema. Hence, you are free to modify the schema if you wish while integrating the new data requirements for the new business. Your group presentation should include a final schema diagram in the form of a Collection Relationship Diagram that accommodates the new business model and data requirements.

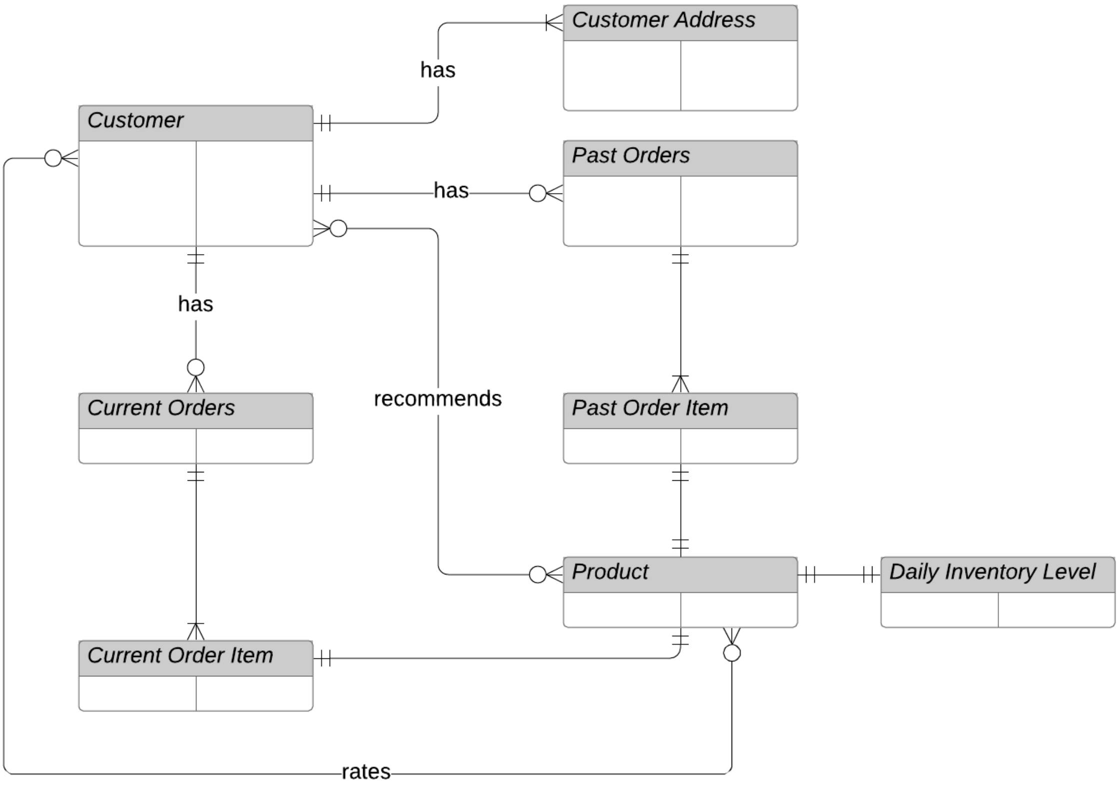

Figure 1: Old Amazone Schema

Requirements, Assumptions, and Projections

1. **Customer** contains attributes such as name, gender, age, customer ID, address, etc. We assume that the customer information is relatively stable. The number of customers is huge (in millions).

2. **Customer Address** has attributes such as house number, street, city, postcode/zip code, etc. A customer can have more than one address (such as billing address(es), shipping address(es), etc.). We assume that the addresses are relatively stable.

3. Customers may have **Past Orders**, each comprising an order ID and the total order cost. Each order is linked with a list of ordered products represented by **Past Order Items**, where each item has an attribute giving its quantity in its order. Each ordered item is linked with one product in **Product**. We assume

that the number of past orders can be large for individual customers (say 200 orders) and is rarely queried.

4. **Current Orders** have a similar structure to **Past Orders**. We assume that the number of current orders of a customer is small (say 5-10 orders), but this information is frequently retrieved.

5. We assume that there are two product segments, **Fresh** and **Others**. **Fresh products** are groceries, which must be picked and delivered from the nearby Morrizon grocery store. Hence, each fresh product has attributes that associate it with a store. Each **Store** has attributes such as an address, location (latitude and longitude), list of grocery items available for ordering and pickup, etc.

6. All fresh orders are picked up by Amazone partners (delivery drivers) registered as **Partners**. Partners' attributes are similar to customer attributes except that we need to know their present location if they are idle or active, and if they are active, we need to know if they are on a delivery errand. These attributes will help the application to decide which partner to assign a pickup and delivery task. Also, we may want to know the details of products delivered and some statistics to calculate pay-outs to each partner – you are to decide these statistics.

7. In a real-world system, there may be thousands of **Fresh Product** categories, but Amazone has decided to start with three **Fresh Product** categories, which are *bakery, drinks, and fruits and vegetables*. Each product type in each of the fresh categories has the following attributes:

   - product name
   - short description
   - product dimensions
   - product weight or quantity (for liquids)
   - expiry date
   - country of origin
   - average customer rating score
   - standard price to customers
   - cost of products from Morrizon (i.e., the cost to Amazone)

8. In a real-world system, there may be thousands of product types for **Other Products**. Here, we assume that there are four types: **books**, **CDs**, **mobile phones,** and **home appliances**. The number of products is huge (millions). Each product type has the following attributes:

- product name
- short description
- product dimensions
- shipping weight
- average customer rating score
- standard price to the customer
- cost of products from the supplier (i.e. cost to Amazone)

Additionally, each product type has its specific attributes.

Book has:

- author name
- publisher
- year of publication
- ISBN

CD has:

- artist name
- number of tracks
- total playing time
- publisher

Mobile phone has:

- brand
- model
- colour
- features

The home appliance has:

- colour
- voltage
- style

9. Each product is linked with its **Daily Inventory Level** for the past five years, with attributes:
    - date
    - inventory quantity
    - storage warehouse location
    - storage warehouse name

10. A customer can give a rating to a product, but not every customer will rate every product. In fact, the rating is typically very sparse. For example, for each product, perhaps only 1% of the customers have ever rated it, and for each customer, they have only rated about 0.1% of the products. The rating information needs to be intensively retrieved to calculate predicted ratings for products a customer hasn't bought yet and will be used to derive a set of recommended products for each customer. Both the rating and the recommended products are frequently updated. The recommended products must be displayed quickly every time a customer logs in to Amazone.

Assignment tasks

Your group must first decide how to split the project into sub-tasks between its members, and your presentation slide **must contain a list of members and sub-tasks assigned to each member**. Each group should choose one member to be the **coordinator**, who will be responsible for submitting all the assignment deliverables.

Task 1: Design a NoSQL schema for Amazone

Use the format below (Collection Relationship) to describe and present your final schema, and include this in your group presentation slide:

```
COLLECTION1 {
field1: type                          // for single valued fields
field2: {
   sub-field1: type                   // for embedded fields
   sub-field2: type
   … etc
}
field3: [ type ]                      // for arrays
field4: ref<COLLECTION.Field>         // for referencing fields
…etc
} // end of COLLECTION1
COLLECTION2 {
… etc
} // end of COLLECTION2
… etc
```

Task 2: Implement a demonstration database

Based on your schema from Task 1, implement a demonstration database using MongoDB Compass or MongoDB Shell. The specific contents (such as the customer names, addresses, product names, etc) are for you to choose, but the following 10 requirements must be satisfied:

1. at least 20 customers.
2. at least 10 product samples for each product (book, CD, mobile phone, and home appliance).
3. at least 5 fresh products from each fresh product category
4. at least 5 store locations for fresh products pickup and delivery
5. at least 5 partners for instant pickup and delivery
6. each customer has at least 2 current orders and at least 5 past orders.

7. each customer has rated at least 3 products; at least 2 customers have rated each product.
8. recommend each customer with at least 2 products.
9. only logs for 2 days of daily inventory levels are needed for each product.
10. at least one of the collections should be indexed for an efficient query. You are to decide which of the collections the index should be created and on which field.

You should implement at least 10 queries designed by yourselves using MongoDB Compass, MongoDB Shell, Python or NodeJS to demonstrate that your database works. You should implement the queries as follows:

1. At least 2 queries indicating a customer ordering a fresh product. The query should include the assignment of pickup and delivery tasks to a partner based on location parameters. The query should return, e.g., details of the product ordered, delivery partner location and/ or ETA, and details of the delivery partner – name and ratings (if available).

2. At least 1 query that indicates a user searching for available fresh products. The products should be displayed based on the user's location.

3. At least 2 queries that indicate a customer ordering a product, adding it to the cart and making payment.

4. At least 2 queries indicating a manager checking sales and inventory performance. Where possible, the query result should be visualised using Pandas (table) or Matplotlib (charts).

5. The remaining queries should be designed by yourself. The designed queries must be realistic, that is, they should be quarries that apply to an ecommerce application. At least 30% of these queries must use the aggregation pipeline.

Export and collate all your collections in JSON format (export from MongoDB Compass to JSON files), including query commands and the corresponding query results (by copying and pasting from MongoDB Compass, MongoDB Shell, Python, or NodeJS).

Deliverables

There are four deliverables:

1 one presentation slide per group
2 one recorded video presentation per group
3 one zipped copy of collections, queries, and results per group
4 one report per individual student.

A2.1 Group Deliverables (70% of the Assignment 2 assessment; each group member receives the same mark). A **presentation slide** and **presentation video** co-designed and presented by all group members, describing how the project was partitioned into sub-tasks between group members; your NoSQL database design; your reasons for your design decisions, design patterns, operations assumptions; and queries design decisions. Zip your database collections as JSON files, including your query commands, and the results of the queries and upload separately. The group presentation slides, the presentation video, and the database and queries zipped files should be submitted on Blackboard by one student on behalf of the group by 15:00 Friday, 15 December 2023.

A2.2 Individual report (30% of the Assignment 2 assessment). Each group member should do the following (1,000 --1,500 words):

1. Write a personal reflection essay in which you discuss the following:

a) your personal role in the Amazone project: what you designed, implemented, and the rationale for the different decisions made.
b) a reflection on your learning experience from the Amazone project: what did you learn? What did you find challenging?

2. The company has decided to expand its operations to some EU countries and has decided to set up an additional data center in Europe for this purpose (you may use a diagram where appropriate).

a) What type of replication algorithm/strategy will you suggest and why?
b) What type of partition and allocation strategy will you suggest and why?
c) Will this operations expansion to Europe require a re-design of the database, and why?

Submit your individual report (clearly marked with your student number) as a **PDF via Blackboard** (in the Assignment 2.2 submission area) by 15:00 Friday, 15 December 2023.

A2.1 Group deliverables marking scheme (total: 20 marks)

a) The presentation slide is formatted and presented clearly and professionally [2] (presentation slides)
(2: professionally presented; 1: acceptably presented; 0: poorly presented)

b) The presentation by the team is coherent, professional, and addressed all key issues presented in the assignment brief [2] (presentation video and presentation slides)
(2: coherent, professional and addressed all key issues; 1: fairly coherent, professional and addressed some key issues; 0: incoherent, unprofessional and addressed a few key issues)

c) Partitioning of tasks between group members [1] (presentation video and presentation slides)
(1: clearly described and evenly partitioned; 0: not clearly described and/or not evenly partitioned)

d) Description of design and implementation [3] (presentation video and presentation slides)
(3: professionally described with all relevant details covered; 2: adequately described with relevant details covered; 1: description is brief with some details omitted; 0: inadequate description)

e) Description of reasons behind design decisions [3] (presentation video and presentation slides)
(3: professionally described with all relevant details covered; 2: adequately described with relevant details covered; 1: description is brief with some details omitted; 0: inadequate description)

f) Sample data implemented [3] (presentation video, presentation slides, and zipped database collection and queries)
(3: implemented fully according to task specification; 2: implemented mostly according to task specification; 1: implemented partially according to task specification; 0: inadequately implemented)

g) Sample queries implemented [3] (presentation video, presentation slides, and zipped database collection and queries)
(3: implemented fully according to task specification; 2: implemented mostly according to task specification; 1: implemented partially according to task specification; 0: inadequately implemented)

h) Query results presented [3] (presentation video, presentation slides, and zipped database collection and queries)
(3: presented fully and results are correct; 2: presented partially and results are correct; 1: presented partially and results are partially correct; 0: inadequately presented and results are not correct)

A2.2 Individual report marking scheme (total: 10 marks)

a) The report is formatted, literate, and presented clearly and professionally [2]
(2: well-written throughout with almost no errors and professionally presented; 1: mostly well-written, but with several errors and acceptably presented; 0: poorly-written with numerous errors and poorly presented)

b) Discussion of personal role in the Amazone project [1]
(1: role is clearly described; 0.5: role is vaguely described; 0: no meaningful description)

c) Reflection on learning experience in the Amazone project [1]
(1: learning and challenges fully described; 0.5: learning and challenges adequately described; 0: no meaningful description)

d) Discussion on the type of replication algorithm/strategy and reasons [2]
(2: insightful discussion demonstrating good understanding; 1: discussion demonstrating fair understanding; 0: discussion is largely irrelevant)

e) Discussion on type of partition, allocation strategy and reasons [2]
(2: insightful discussion demonstrating good understanding; 1: discussion demonstrating fair understanding; 0: discussion is largely irrelevant)

f) Discussion on the implication of European expansion on the database design [2]
(2: insightful discussion demonstrating good understanding; 1: discussion demonstrating fair understanding; 0: discussion is largely irrelevant)

[**ends**]