MANCHESTER 1824
The University of Manchester

Understanding Database

Group 21

yifei.huang-3@postgrad.manchester.ac.uk [ID11597876]
huilin.he@postgrad.manchester.ac.uk [ID11531666]
yinhang.zhang@postgrad.manchester.ac.uk [ID11613159]
jingzhou.shi@postgrad.manchester.ac.uk [ID14104388]
zihan.zhang-21@postgrad.manchester.ac.uk [ID14142055
rahmah.rifat@postgrad.manchester.ac.uk [ID10534099]
tania.diamanta@postgrad.manchester.ac.uk [ID11585796]

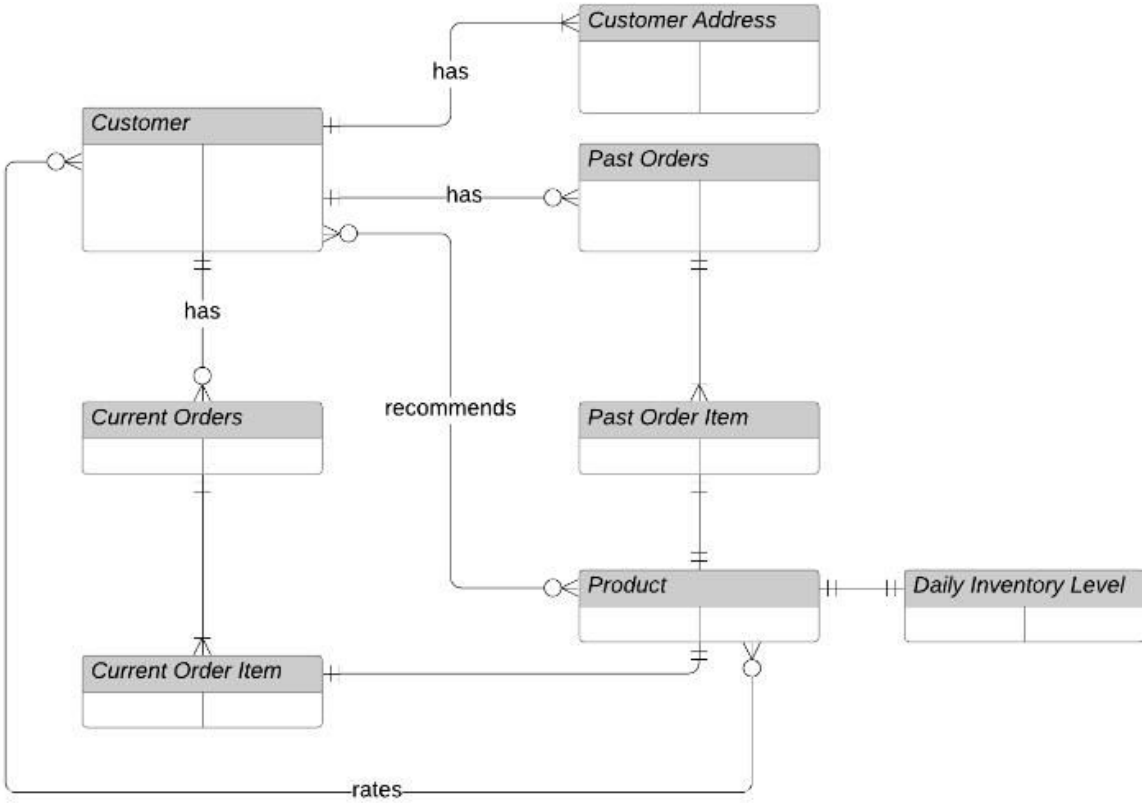
MANCHESTER 1824
The University of Manchester

Deliverables

- **Task allocation between group members;**
- **NoSQL database design showcase**
- **Design decisions, design patterns, operations assumptions**
- **Queries design decisions.**

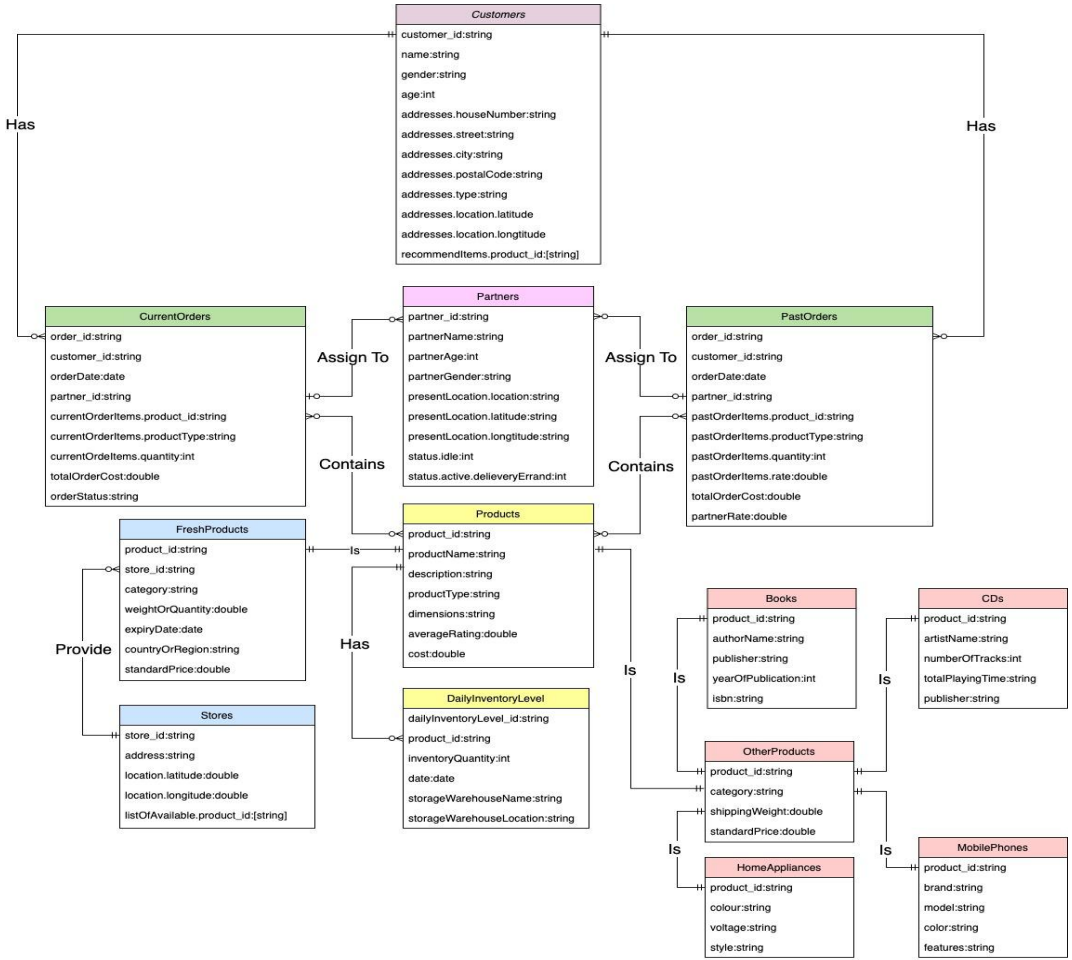
MANCHESTER 1824
The University of Manchester

[Original] NoSQL database design



MANCHESTER 1824
The University of Manchester

[Updated] NoSQL database design



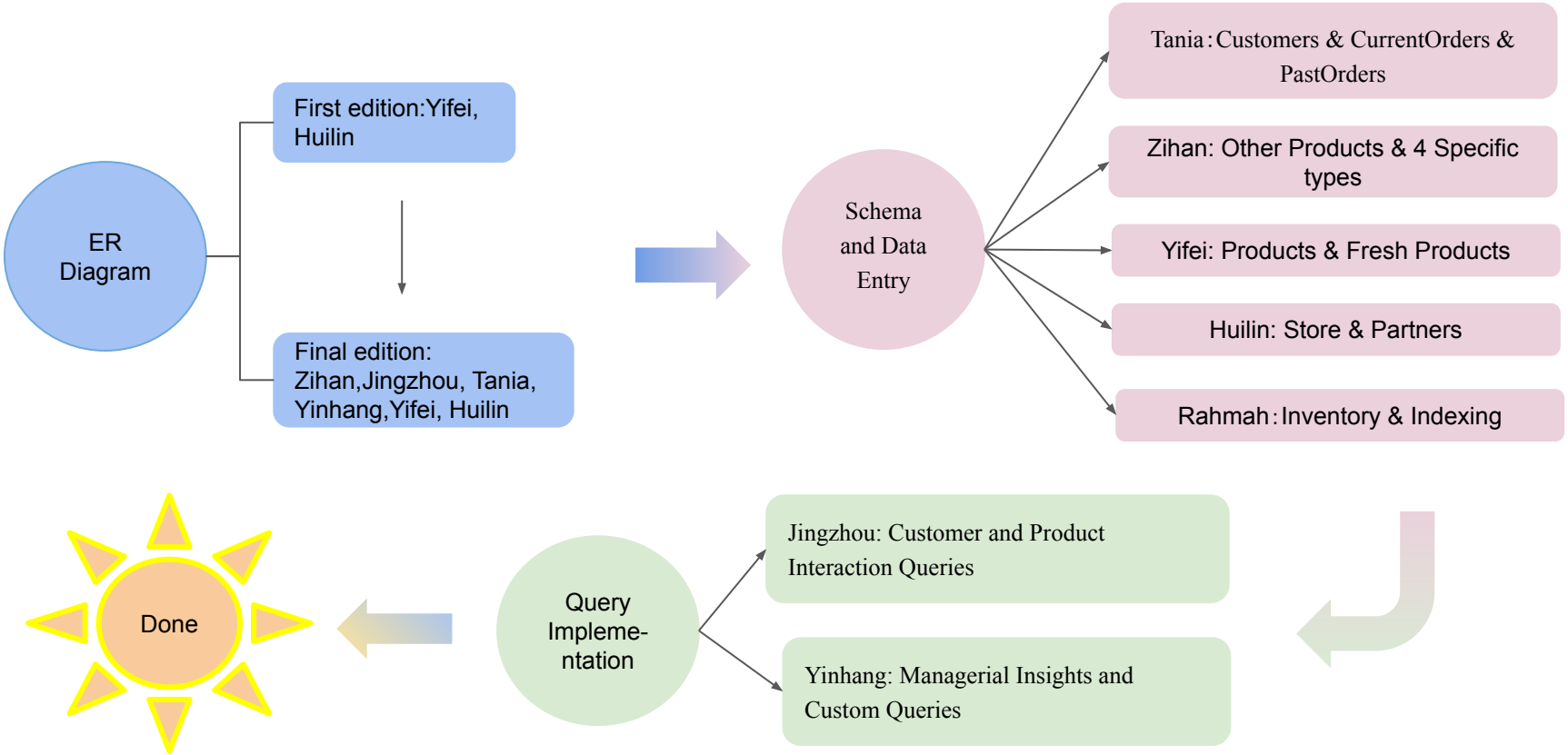
MANCHESTER 1824
The University of Manchester

Why Document Format?

01	**Flexibility**	• Variety of products • Unique and tailored structure per collection • Easy to update
02	**Simplicity**	• Nested data in a single document • I.e. Orders (multiple products per order)
03	**Speed**	• No joins • Faster, cheaper, and more simple queries
04	**Scalability**	• Horizontal • Ideal for high traffic

MANCHESTER 1824
The University of Manchester

Task Allocation



MANCHESTER 1824
The University of Manchester

Operations assumptions

- Coordinates (In GeoJSON format) is added for easy location management
- The distance we used are straight line distance, but in reality this is not the case
- Fresh product are stored in the store and other products are placed in the warehouse

MANCHESTER 1824
The University of Manchester

Design decisions

- Combine collections
- Separating ‘FreshProducts’ & ‘OtherProducts’, all following ‘Products’
- Adding longitude and latitude to ‘Customers’ and ‘Stores’ to calculate distance
- Add orderStatus to ‘CurrentOrders’
- Add averageRating to ‘Products’; Calculate the average from the rate of ‘PastordeItems’

MANCHESTER 1824
The University of Manchester

Design Patterns - 1

What are Design Patterns?

- Solutions to database design problems that are reusable
- Ensure the database is efficient and prevents redundancy, ensuring it can be scaled to other requirements. For example, including all products in Products would make it harder to find specific products from the database.

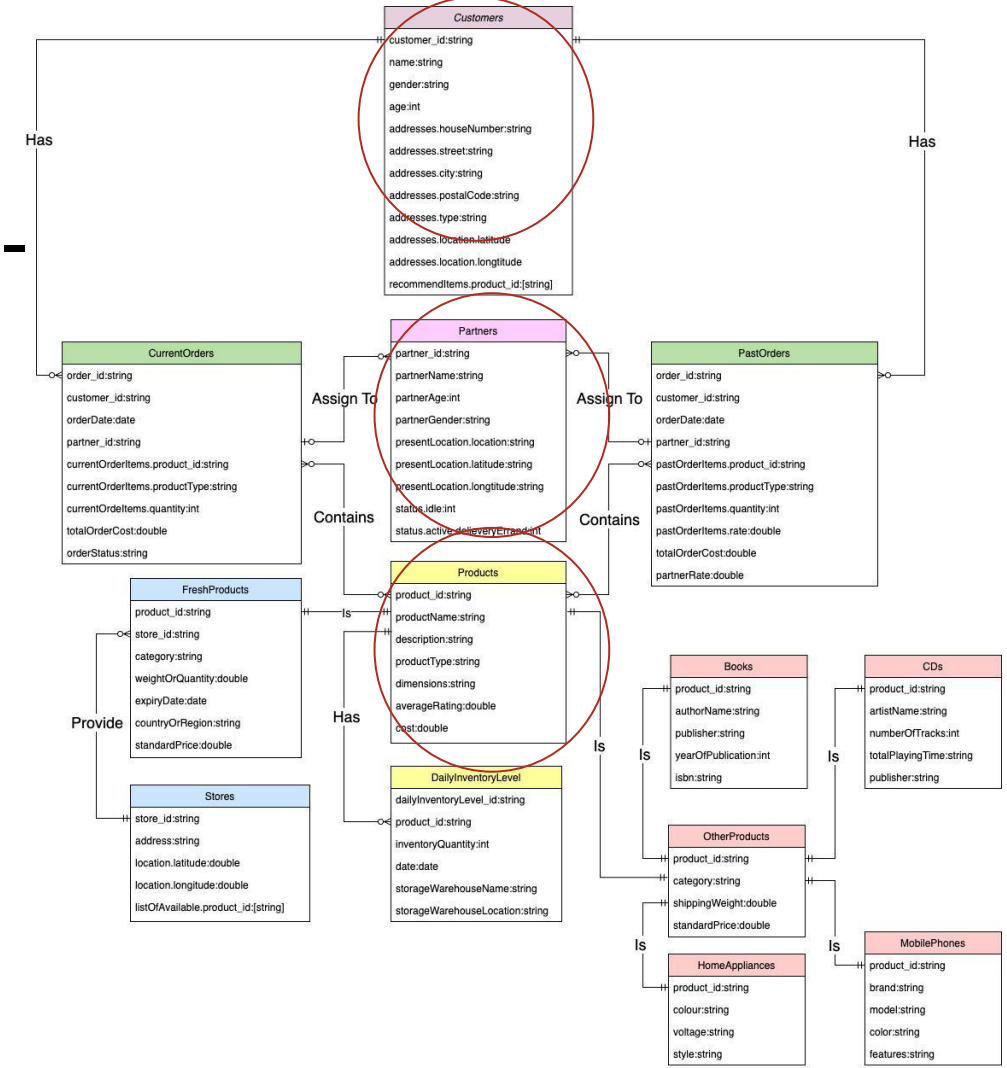
Reasons to use design patterns in this Database:

- Handling diverse product attributes
- Integrate spatial data for entities like Store, where location-based attributes exist
- Enhance relationships between entities, such as: customers, orders; products, stores.

MANCHESTER 1824
The University of Manchester

Design Patterns - ER Diagram

Key entities highlighted in ER diagram:

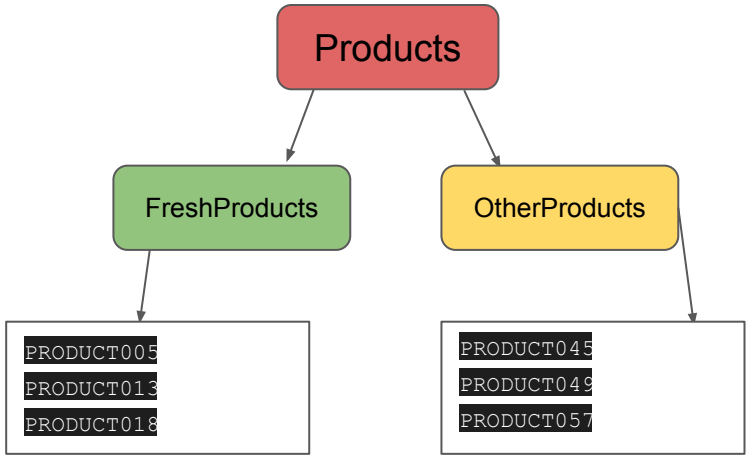


Key Design Patterns - 3

- **Entity-Attribute-Value model:**

The database separates Product attributes into child tables like:

- CDs
- Books
- FreshProducts etc.



Where Books, CDs, MobilePhones and HomeAppliances belong to OtherProducts, which is then connected to Products.

This helps to manage product variations efficiently; as some entities have less/more attributes. This key-value pattern stores each data item as a pair of unique keys and values. General product attributes exist in Products and more specific attributes in each child entity. Here, these entities inherit attributes through ‘product_id.’

Key Design Patterns - 4

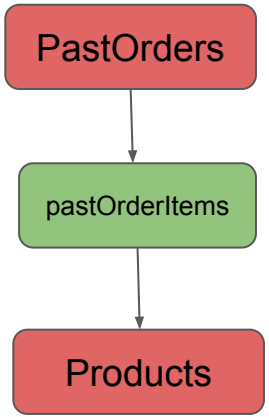
- **Parent-Child design pattern:**

This design allows orders to contain multiple items in a single entity, which in turn, reduces redundancy as there are less repeated attributes.

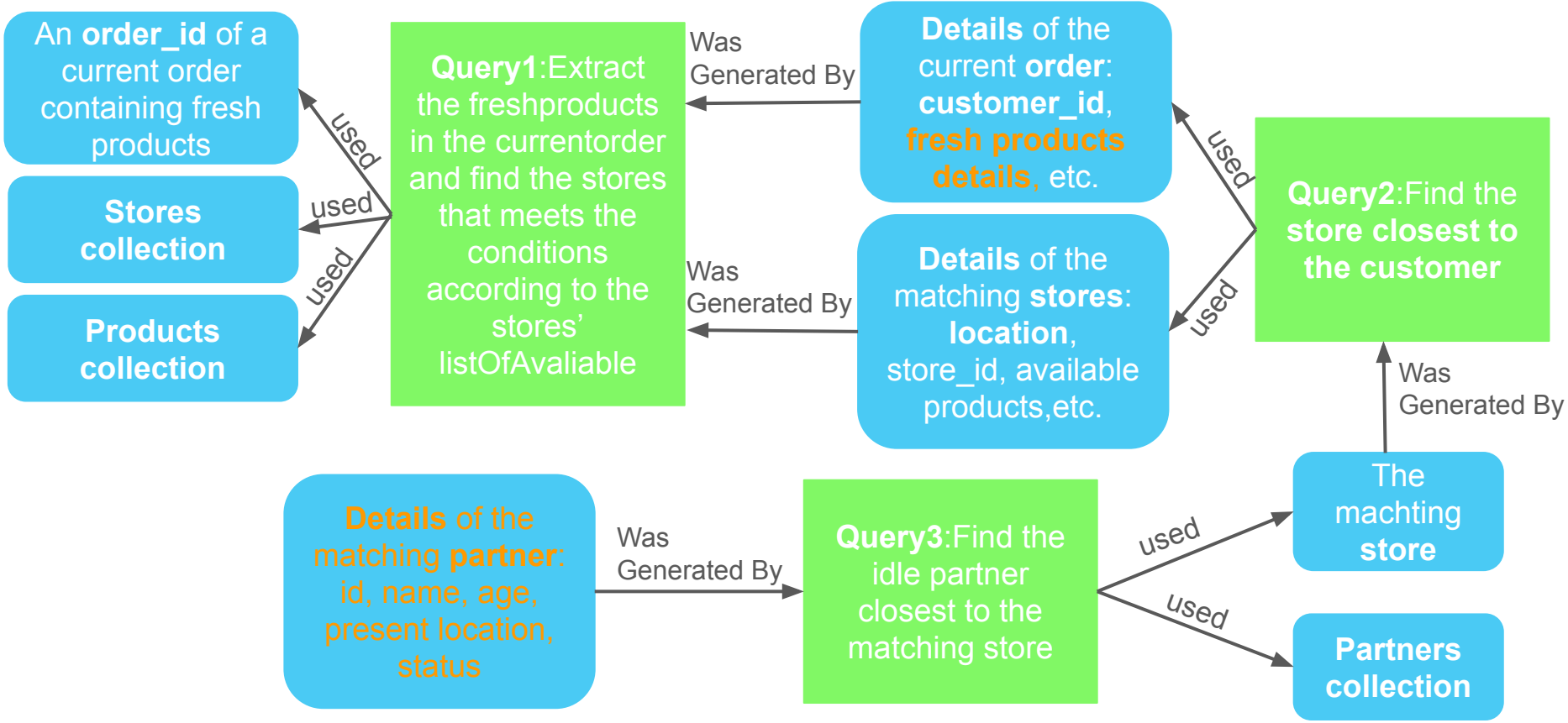
Example:

PastOrders and CurrentOrders contain an array of items which include product_id, productType and quantity etc.

This allows for efficient order processing.

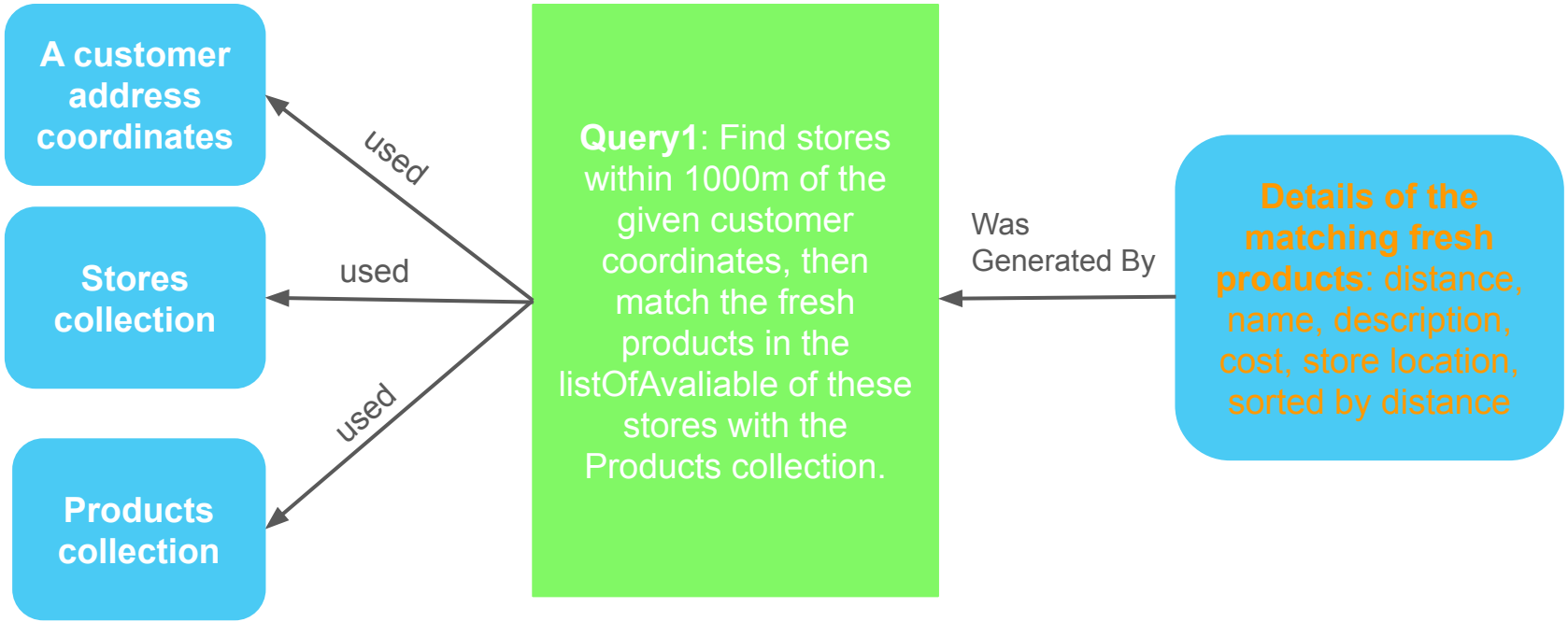


Queries design decisions — Task 1

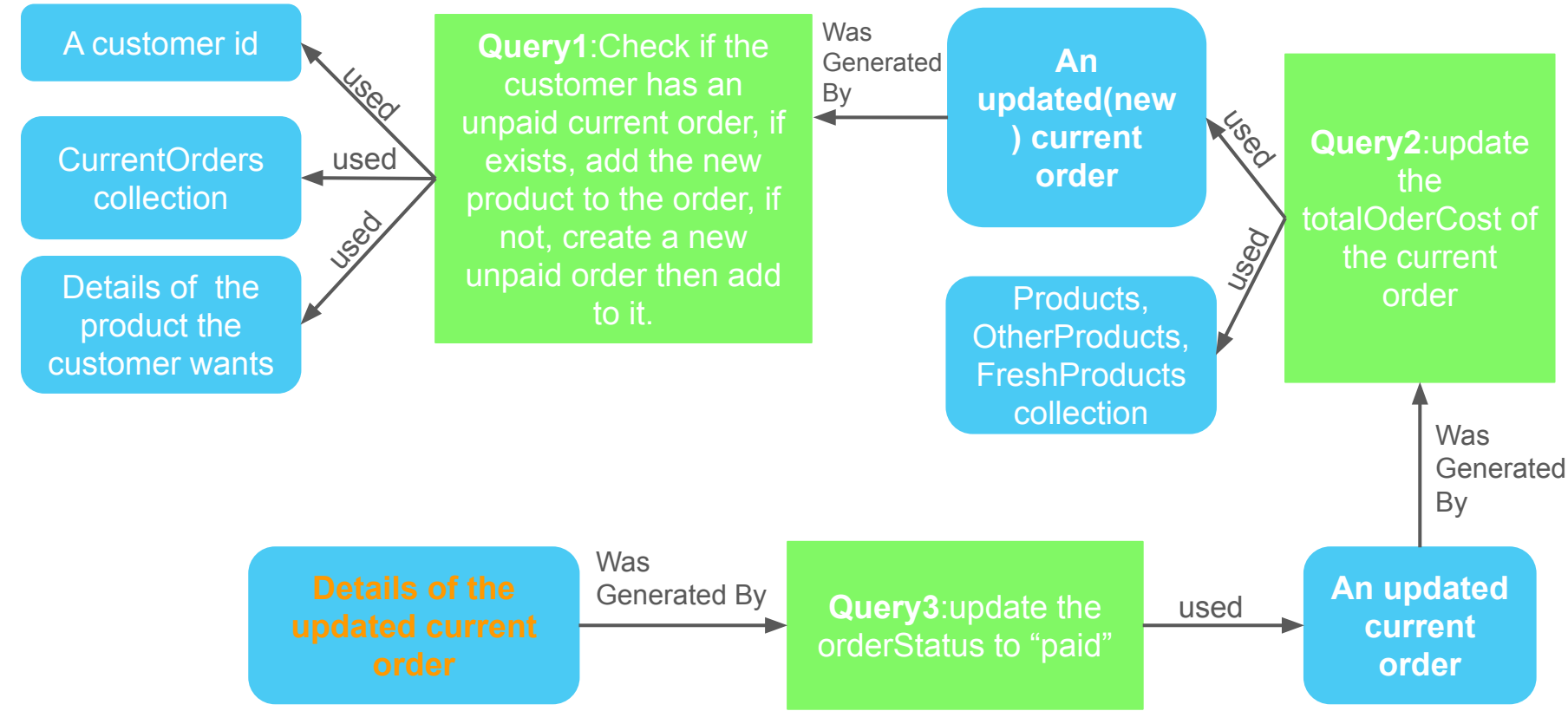


MANCHESTER 1824
The University of Manchester

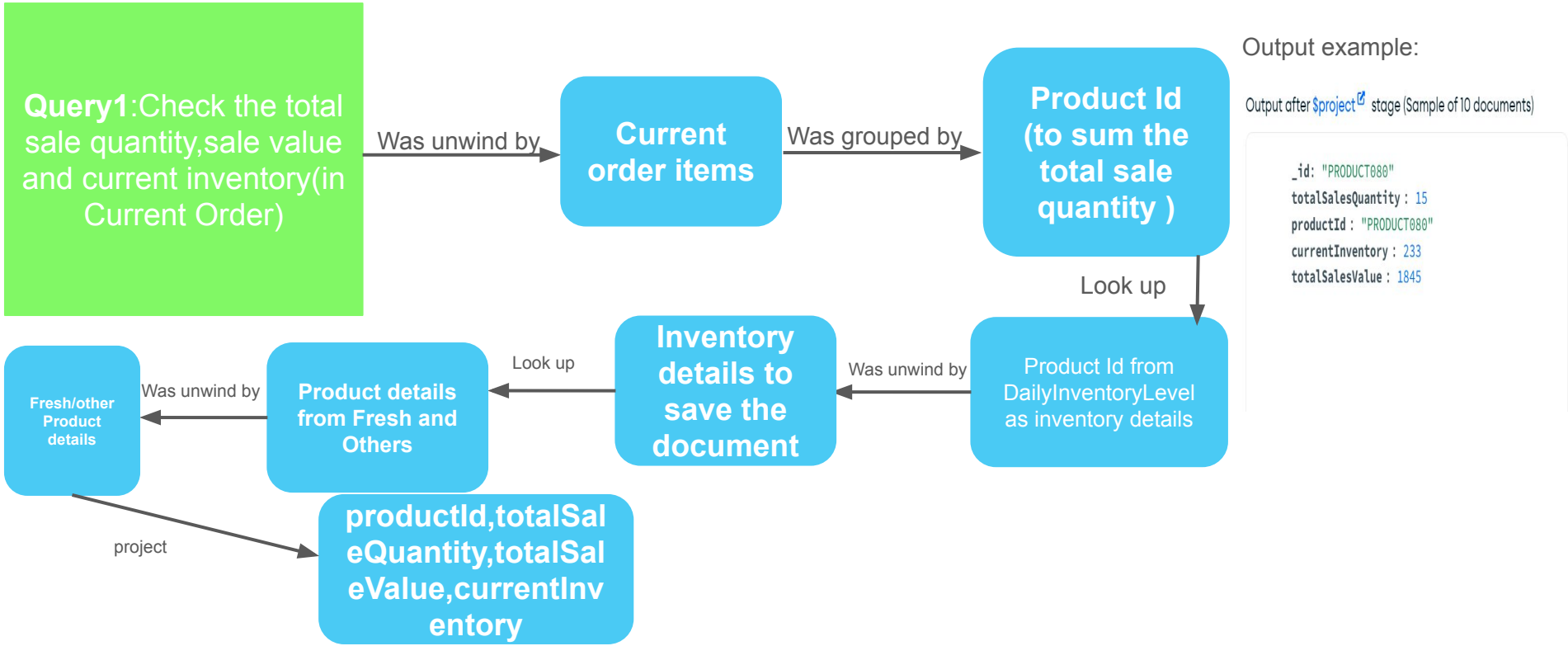
Queries design decisions — Task 2



Queries design decisions — Task 3

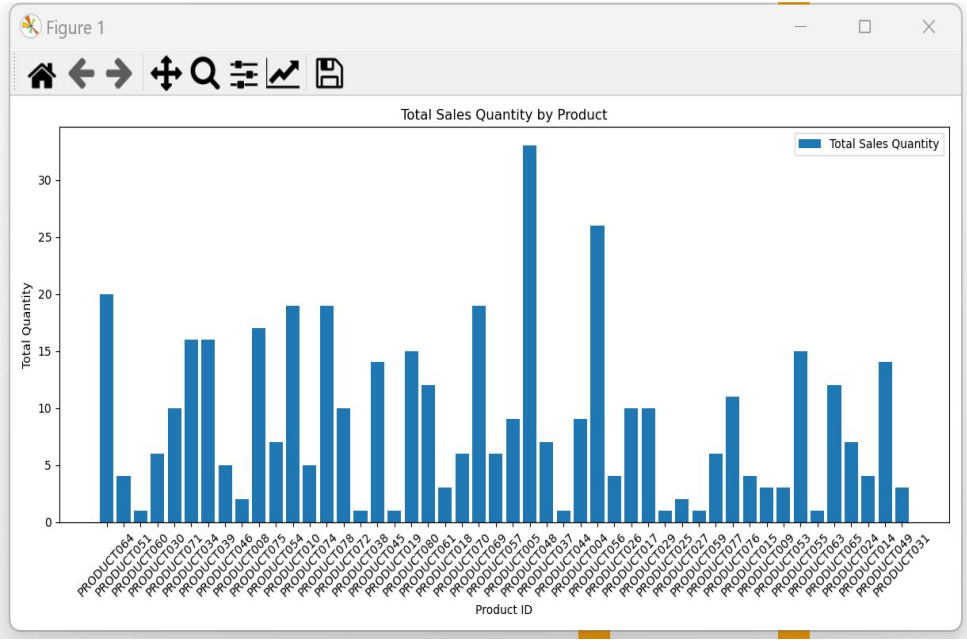


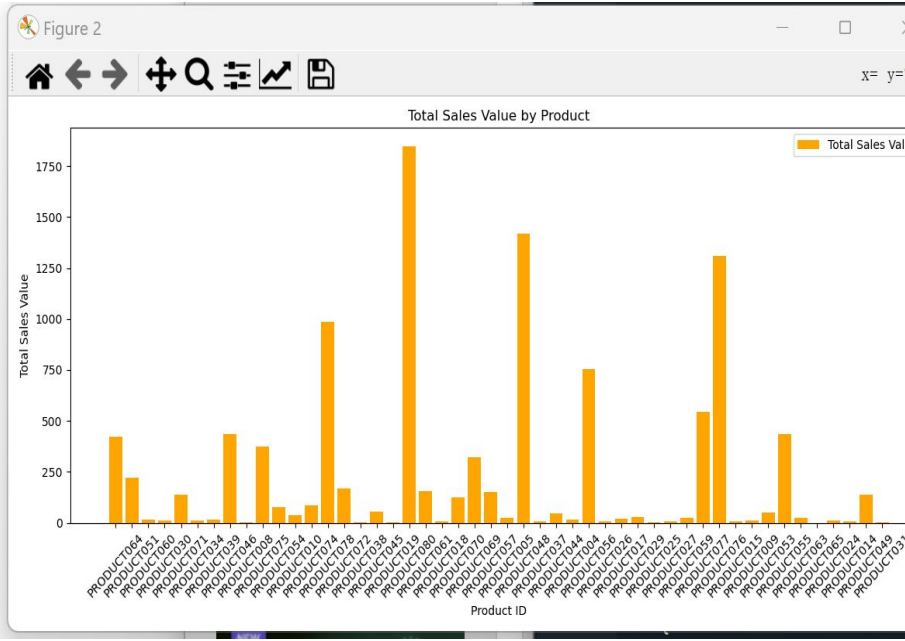
Queries design decisions — Task 4



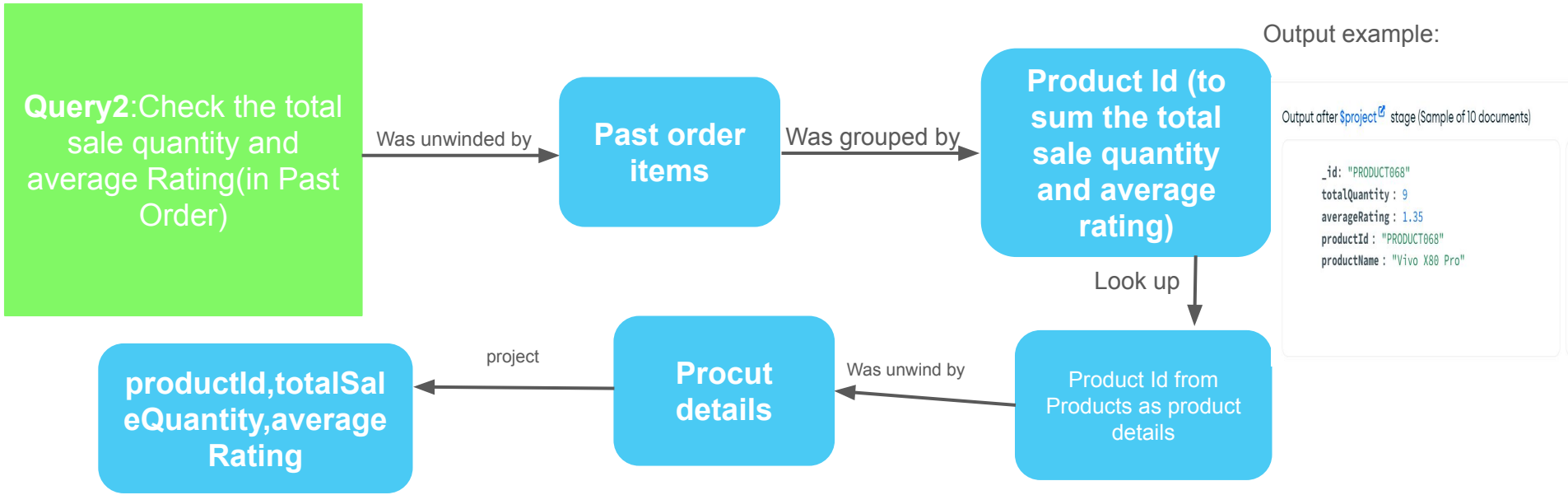
MANCHESTER 1824
The University of Manchester

Visualization



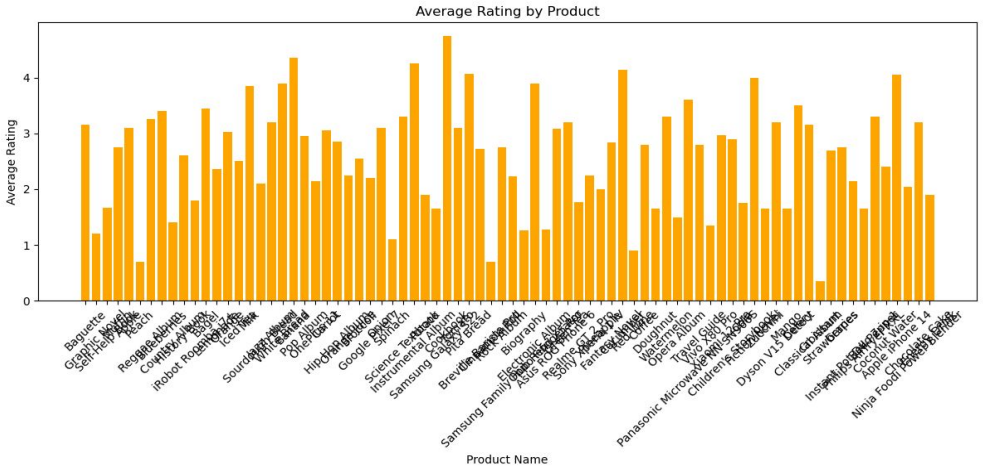


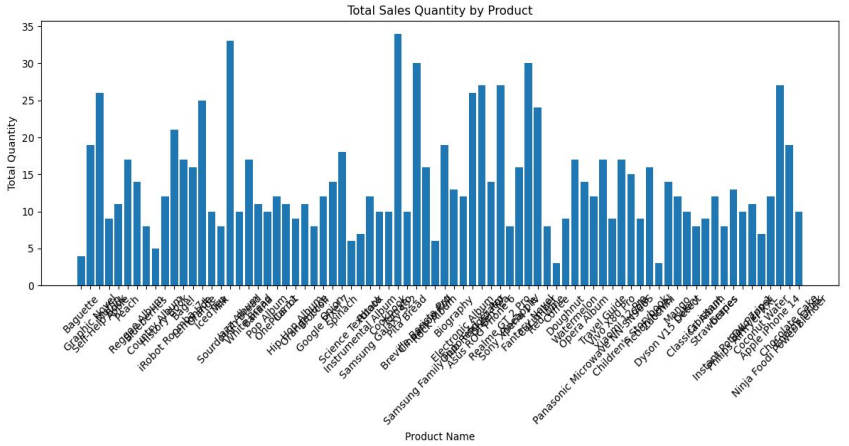
Queries design decisions — Task 4



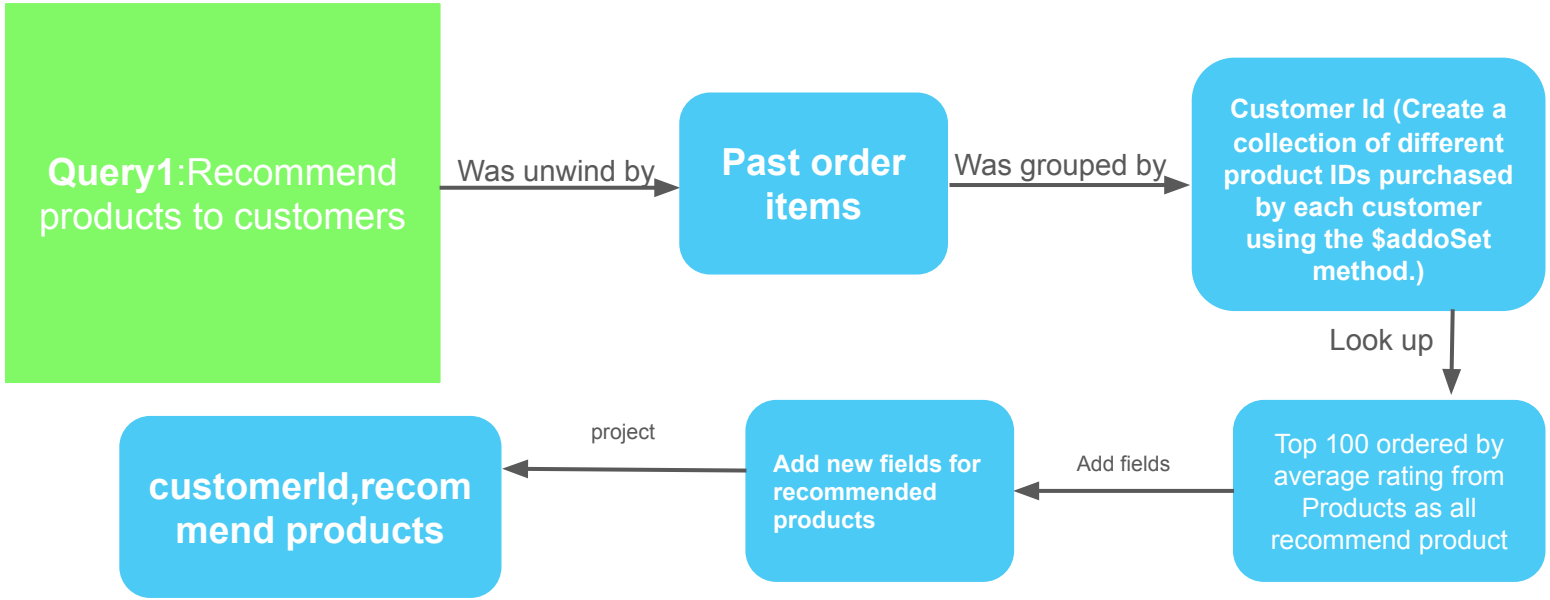
MANCHESTER 1824
The University of Manchester

Visualization



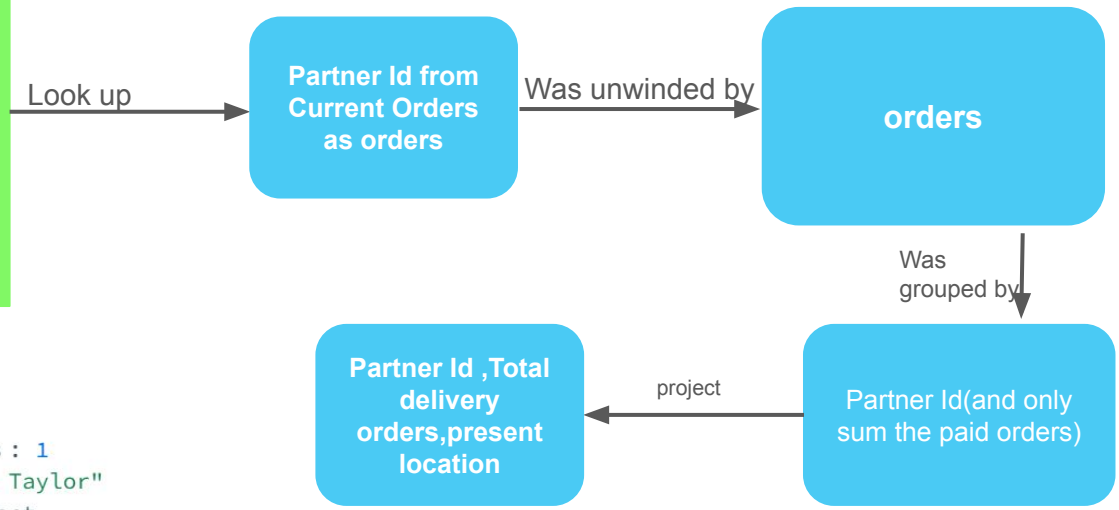


Queries design decisions — Task 5



Queries design decisions — Task 5

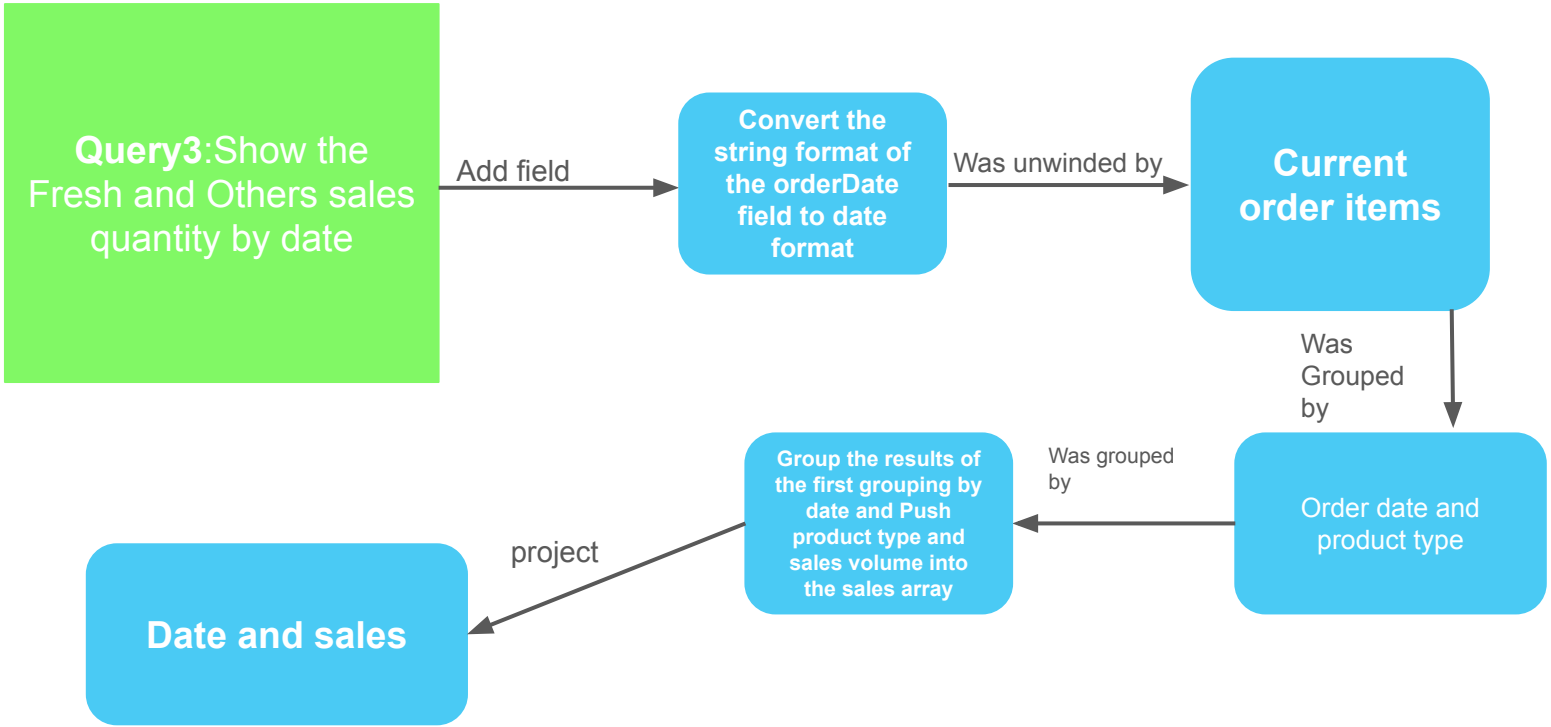
Query2:Show the current location and number of orders for each partners



Output example:


```
_id: "PARTNER002"
Total_delivery_Orders : 1
partnerName : "Sophia Taylor"
▾ presentLocation : Object
    type : "Point"
  ▾ coordinates : Array (2)
      0: -2.2483
      1: 53.4771
partner_id : "PARTNER002"
```

Queries design decisions — Task 5



MANCHESTER 1824
The University of Manchester

Visualization