Otabek
Rahmonov

Otabek Rahmonov superstore

Business area

Overview

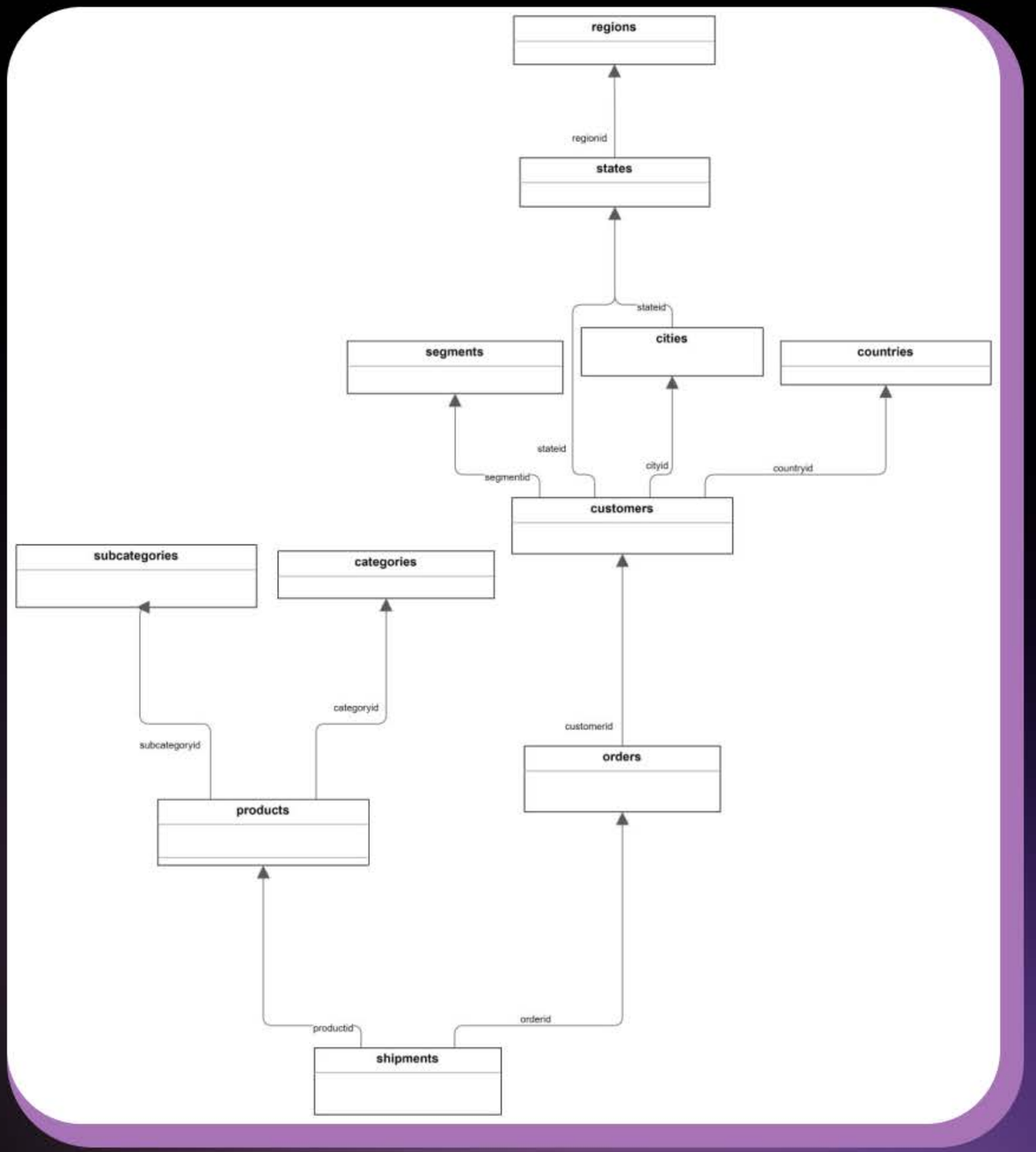
The business area for the "Superstore" database revolves around retail operations and sales management

The dataset contains tables representing various aspects of a retail business, including shipments, products, orders, customers, and geographical entities such as cities, countries, states, and regions. Additionally, the database includes tables for segments, subcategories, and categories, reflecting the product categorization and segmentation strategies employed by the business. Overall, the database serves as a comprehensive repository for managing and analyzing retail sales data, supporting functions such as inventory management, order processing, customer relationship management, and geographical analytics.

Part 1

Overview of tables and columns

I created tables for segments, countries, regions, states, cities, categories, subcategories, products, customers, orders, and shipments, adhering to the requirements of 3NF. I have established primary keys for each table and appropriate foreign key relationships to maintain data integrity.



Relations

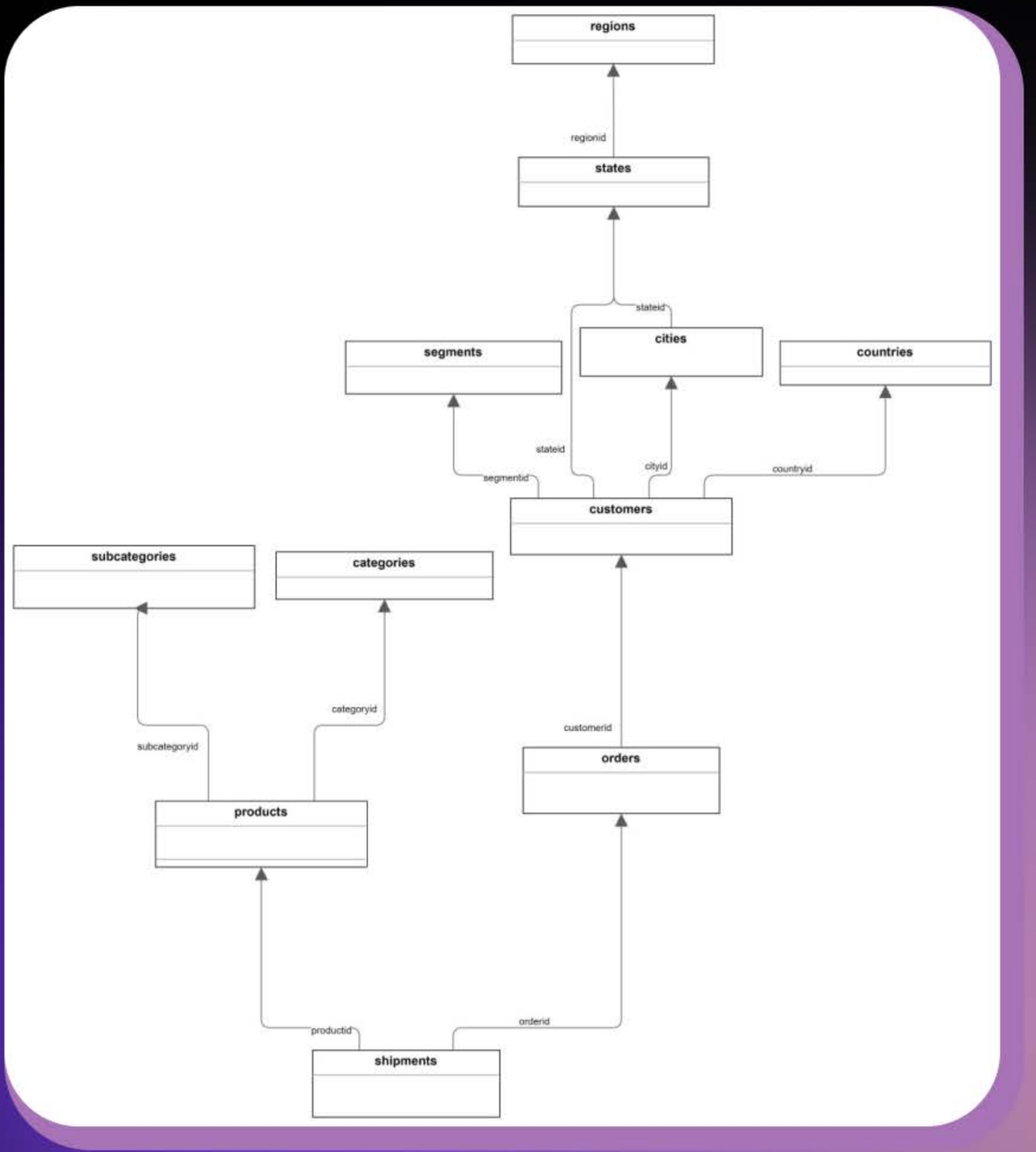
Segments - Customers:
- One-to-Many Relationship

Countries - States:
- One-to-Many Relationship

Regions - States:
- One-to-Many Relationship

States - Cities:
- One-to-Many Relationship

Categories - Subcategories:
- One-to-Many Relationship

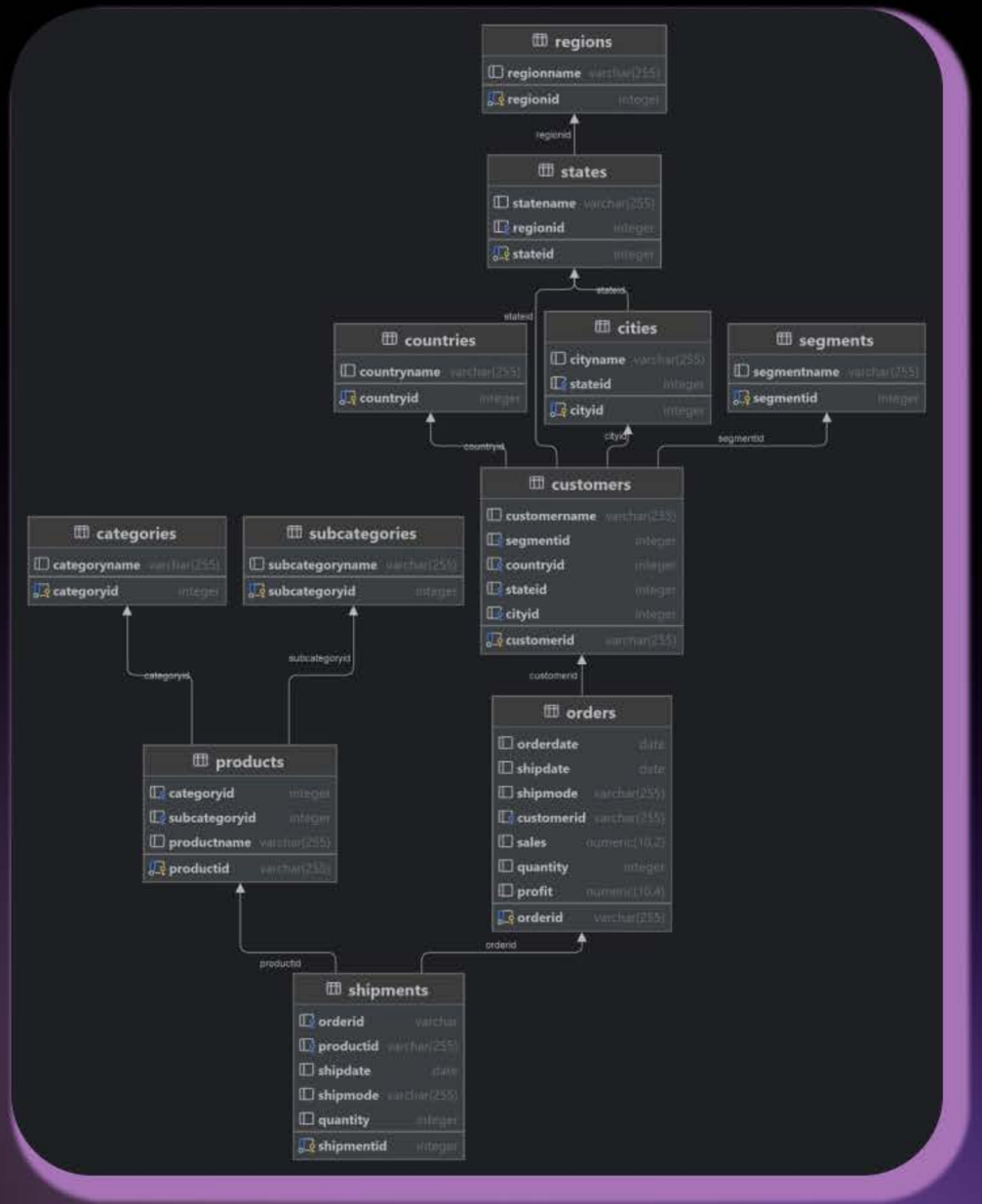


Populating the tables

Populating Tables with Data Samples
To ensure that our database is fully functional and ready for use, we populated the tables with data samples from our chosen dataset. This process involved creating DML (Data Manipulation Language) scripts to insert data into each table.

Outcome:

- Our tables are now populated with representative data samples, allowing us to test the functionality of our database and perform various queries and analyses.



Functions and views

As part of ensuring the flexibility and usability of our database, I have implemented a custom function for updating data in my tables. This function allows me to modify the value of a specific column in a specified row, providing a streamlined approach to data management.

And My view is designed to extract and present essential analytics pertaining to the most recent quarter's data, ensuring relevance and accuracy.

Problems Faced

Programs errors and also when uploading a dataset, dealing with nulls and duplicates

Part 2

Creating a Denormalized Physical Database

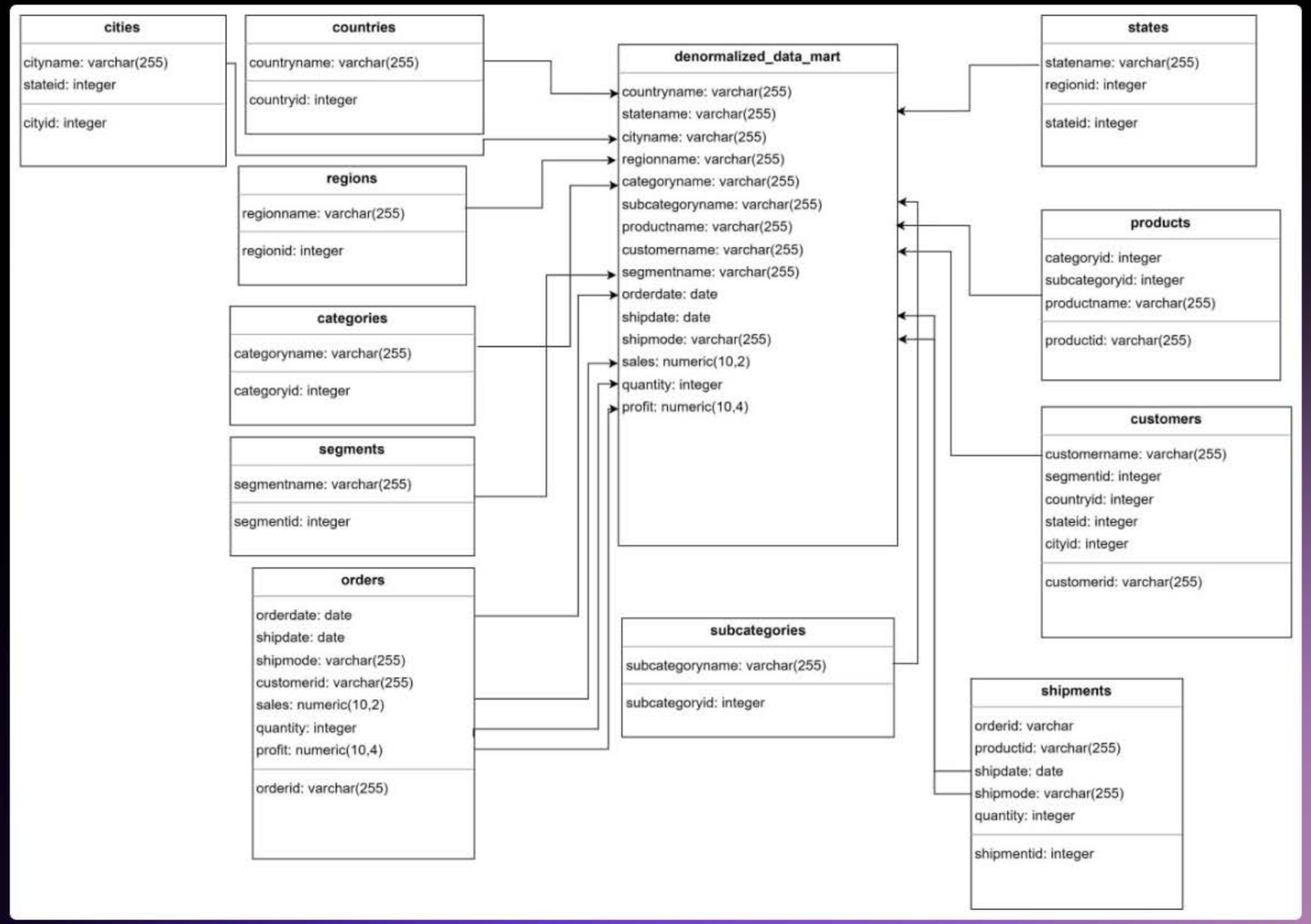
I have established a denormalized physical database with a separate schema, building upon the foundation of our normalized database structure. This denormalized layer consolidates and simplifies data across multiple tables, optimizing query performance and facilitating business analytics.

Data Population:

- We have populated the denormalized tables with data samples generated from our 3NF layer, ensuring consistency and accuracy in the migrated data.
- DML scripts were carefully crafted to insert data into the denormalized tables while adhering to all constraints and business rules established in the normalized layer.

Star Schema

A star schema is a type of data warehouse schema architecture widely used for organizing and representing multidimensional data in a structured format. It consists of one or more fact tables referencing multiple dimension tables. The central concept of a star schema is to divide data into facts and dimensions for efficient querying and analysis.



Business Questions

```
-- What is the distribution of orders across different cities,
-- and what is the count of orders for each city?

SELECT
   cityname,
   order_count
FROM
   (SELECT
      cityname,
      COUNT(*) OVER(PARTITION BY cityname) AS order_count,
      ROW_NUMBER() OVER(PARTITION BY cityname ORDER BY orderdate) AS row_num
    FROM
      data_mart.denormalized_data_mart) AS sub_query
WHERE
      row_num = 1;
```

```
-- Which product has the highest number of orders?

SELECT
   productname,
   COUNT(*) AS order_count
FROM
   data_mart.denormalized_data_mart
GROUP BY
   productname
ORDER BY
   order_count DESC
LIMIT 1;
```

Business Questions

```
-- Who are the top 10 customers based on total sales?

SELECT d.customername, SUM(d.sales) AS total_sales
FROM data_mart.denormalized_data_mart d
GROUP BY d.customername
ORDER BY total_sales DESC
LIMIT 10;
```

```
-- To calculate profit margin for each product:

SELECT d.productname, (SUM(d.profit) / SUM(d.sales)) * 100 AS profit_margin_percentage
FROM data_mart.denormalized_data_mart d
GROUP BY d.productname;
```

Conclusion

Thank You!