DATABASE FOR COUNTERFEIT PRODUCT MANAGEMENT

Name: Durga Naga Padmasree Sappa

ISYS 864 - 01 DATA MANAGEMENT FOR ANALYTICS Fall 2024

TABLE OF CONTENTS

INTRODUCTION

Amazon:

Sector: E-commerce / Consumer Discretionary

The organisation chosen for this project is Amazon, a global leader in e-commerce within the consumer discretionary sector. Amazon specializes in online retail, cloud computing, digital streaming, and artificial intelligence. This project focuses on Amazon's Brand Protection initiative, which combats counterfeit products by ensuring product authenticity throughout the supply chain.

The Amazon Transparency Program, a key component of this initiative, assigns unique transparency codes to legitimate product units sold on its platform. These codes enable Amazon and its customers to verify product authenticity, identifying and removing counterfeit products. The database designed in this project serves as the backbone for managing the entire lifecycle of brand protection. It includes:

- Brand registrations to enrol legitimate businesses.
- Product authenticity tracking using transparency codes.
- Customer complaint management related to counterfeit products.
- Monitoring investigations and enforcement actions against fraudulent sellers or products.

OVERVIEW

Area of Focus:

Counterfeit products pose significant risks to consumers and brands in e-commerce, creating a database that tracks and manages the entire lifecycle of product authenticity offers valuable insights that allow us to mitigate this issue. The database for this project will focus on managing brand registrations, tracking product authenticity, handling customer complaints regarding counterfeit products, and monitoring investigations and enforcement actions taken against sellers or products found to be counterfeit. The database will serve as the core for managing all aspects of the brand protection program, from product registration to customer complaints, investigations, and enforcement actions.

Why did we pick this organization

Our group is interested in the e-commerce industry and aspires to build careers. We chose Amazon as our project focus due to its global dominance and industry-leading practices. By working with Amazon's data, we aim to gain practical experience in database management, entity relationship modelling, and the underlying logic that drives its unparalleled success. Additionally, one of our team members has prior experience working at Amazon, which will provide us with valuable insights and direction. Her firsthand knowledge will deepen our understanding of Amazon's operations and significantly contribute to the success of our data management project.

How you obtain the data

The dataset for this project was meticulously created using Mockaroo, a data generation tool, and a combination of manual modifications to align with the project's requirements. The process involved fabricating data to simulate realistic scenarios while ensuring the dataset supported the project's objectives.

We started using Mockaroo to generate initial datasets for the primary entities, such as brands, products, customers, complaints, investigations, and enforcement actions. The generated data included plausible values for attributes like dates, product categories, and numerical fields (e.g., units sold and complaint counts). These datasets were designed to reflect real-world scenarios while avoiding reliance on sensitive or proprietary data.

DATA PREPROCESSING

Data Cleaning and Preparation

1. Alignment with Topic Objectives:
    - The data was reviewed to align with our focus on counterfeit product tracking, complaint management, and enforcement actions. Any discrepancies were adjusted to match the conceptual goals of the project.
    - Super and sub-type tables were introduced, resulting in 14 interconnected tables representing comprehensive relationships among entities like products, their categories or enforcement actions, and their types (e.g., legal or operational).

2. Data Tweaking for Calculations:
    - Values were selectively tweaked to facilitate meaningful calculations and queries, such as ensuring complaint rates or enforcement outcomes could be analysed effectively.
    - Specific cases were added to highlight patterns in counterfeit complaints, product categories prone to issues, and enforcement success rates.
3. Manual Adjustments Using Excel:
    - Using Excel and formulas, we refined the generated data, ensuring logical consistency and integrity. For instance:
        - Unique relationships between products and brands.
        - Valid references for foreign keys, ensuring proper table linkage.
        - Realistic distribution of complaint and enforcement data to support SQL queries.
4. SQL Data Insertion:
    - Insert statements were crafted using Excel formulas, simplifying the creation of SQL-ready data entries.
    - These statements were pasted directly into the SQL scripts to populate the database tables.

ENTITIES & ATTRIBUTES

The following are short descriptions of each entity:

Tables	Description
1. Brand	The Brand entity tracks information about different brands. A unique brand ID identifies each brand. It contains details such as the brand name, the date it was registered, the contact person's name, email, and phone number for communication. It also includes the brand's current status and the number of products associated with the brand.
2. Product	The Product entity holds data on the products offered by various brands. Each product has a unique product ID. The entity captures information such as the product name, a transparency code for tracking, the product category with Electronics, Clothing, and Furniture as its subtypes, the launch date, the

	product's current status, the number of units sold, and the number of customer complaints received.
3. Customer	The Customer entity manages customer-related information. Each customer is uniquely identified by a customer ID. The entity records the customer's name, email, phone number, address, purchase history (which is a multivalued attribute), and the date of their registration.
4. Complaint	The Complaint entity stores information about customer complaints related to products. Each complaint has a unique complaint ID. It tracks the date the complaint was lodged, a description of the complaint, the current status of the complaint, and the date the complaint was resolved.
5. Investigation	The Investigation entity manages details of investigations initiated in response to complaints. Each investigation is identified by an investigation ID. This entity records the start date of the investigation, the person assigned to it, the investigation status, the findings during the investigation, and its conclusion.
6. Enforcement Action	The Enforcement Action entity tracks actions taken as a result of investigations. Each enforcement action has a unique action ID. It captures the type of action taken with subtypes such as Legal action and Operational action, the date of the action, the status of the action, and the outcome of the enforcement.
7. Transaction	The Transaction entity captures details of sales transactions between customers and products. Each transaction has a unique transaction ID. This entity records the date of the transaction, the quantity purchased, the price per unit, and the total amount of the transaction.

Information about their domain constraints

1. Brand Table

Attribute	Data type	Constraints
Brand ID (PK)	Integer	Must be unique, nonnull
Brand Name	String (Varchar)	Length limit (e.g., 100 characters),
Registered Date	Date	Valid date format (YYYYMMDD), nonnull
Contact Person:	String (Varchar)	Length limit (e.g., 100 characters), nonnull
Email	String (Varchar)	Valid email format (e.g., `example@domain.com`), length limit (e.g., 255 characters), nonnull
Phone Number	Integer	Valid phone number format (e.g., country code + number), length limit (e.g., 15 characters), nonnull
Brand Status	String (Varchar)	Must be one of the predefined statuses, nonnull
Number of Products	Integer	Nonnegative, default to 0, nonnull

2. Product Table

Attribute	Data type	Constraints
Product ID (PK)	Integer	Must be unique, nonnull
Product Name	String (Varchar)	Length limit (e.g., 150 characters), nonnull
Transparency Code	String (Varchar)	Unique code format, length limit (e.g., 20 characters), nonnull
Product Category	String (Varchar)	Length limit (e.g., 50 characters), nonnull

Launch Date	Date	Valid date format (YYYYMMDD), nonnull
Status	String (Varchar)	Must be one of the predefined statuses, nonnull
Units Sold	Integer	Nonnegative, default to 0, nonnull
Number of Complaints	Integer	Nonnegative, default to 0, nonnull

3. Customer Table

Attribute	**Data type**	**Constraints**
Customer ID (PK)	Integer	Must be unique, nonnull
Customer Name	String (Varchar)	Length limit (e.g., 150 characters), nonnull
Email	String (Varchar)	Valid email format, length limit (e.g., 255 characters), nonnull
Phone Number	Integer	Valid phone number format, length limit (e.g., 15 characters), nonnull
Address	String (Varchar)	Length limit (e.g., 255 characters), nonnull
Purchase History	Text	Valid JSON format or text, can be null
Date of Registration	Date	Valid date format (YYYYMMDD), nonnull

4. Complaint Table

Attribute	**Data type**	**Constraints**
Complaint ID (PK)	Integer	Must be unique, nonnull
Complaint Date	Date	Valid date format (YYYYMMDD), nonnull

Complaint Description	Text	No specific length limit, nonnull
Complaint Status	String (Varchar)	Must be one of the predefined statuses, nonnull
Resolution Date	Date	Valid date format, can be null

5. Investigation Table

Attribute	**Data type**	**Constraints**
Investigation ID (PK)	Integer	Must be unique, nonnull
Investigation Start Date	Date	Valid date format (YYYYMMDD), nonnull
Assigned Person	String (Varchar)	Length limit (e.g., 100 characters), nonnull
Investigation Status	String (Varchar)	Must be one of the predefined statuses, nonnull
Findings	Text	No specific length limit, can be null
Conclusion	Text	No specific length limit, can be null

6. Enforcement Action Table

Attribute	**Data type**	**Constraints**
Action ID (PK)	Integer	Must be unique, nonnull
Action Type	String (Varchar)	Length limit (e.g., 50 characters), nonnull
Action Date	Date	Valid date format (YYYYMMDD), nonnull
Action Status	String (Varchar)	Must be one of the predefined statuses, nonnull

Outcome	Text	No specific length limit, can be null

7. Transaction Table

Attribute	Data type	Constraints
Transaction ID(PK)	Integer	Must be unique, nonnull
Transaction Date	Date	Valid date format (YYYYMMDD), nonnull
Quantity	Integer	Nonnegative, nonnull
Price	Decimal	Nonnegative, nonnull
Total Amount	Decimal	Nonnegative

Subtype Table

1. Electronic (subtype of Product)

Attribute	Data type	Constraints
Warranty_Period	Integer	representing months or years, non-negative.
Model_Number	String(Varchar)	limited to 50 characters

2. Clothing (subtype of Product)

Attribute	Data type	Constraints
Size	String(Varchar)	values could be 'S', 'M', 'L', 'XL',
Color	String(Varchar)	limited to 50 characters

3. Furniture (subtype of Product)

Attribute	Data type	Constraints

Material	String(Varchar)	limited to 100 characters
Weight	Decimal	precision up to 2 decimal places

4. Legal Action (subtype of Enforcement Action)

Attribute	**Data type**	**Constraints**
Penalty_Amount	Decimal	precision up to 2 decimal places
Legal_Warning	String(Varchar)	limited to 100 characters
Court_Involvement	Boolean	indicates whether legal proceedings are involved

5. Operational Action (subtype of Enforcement Action)

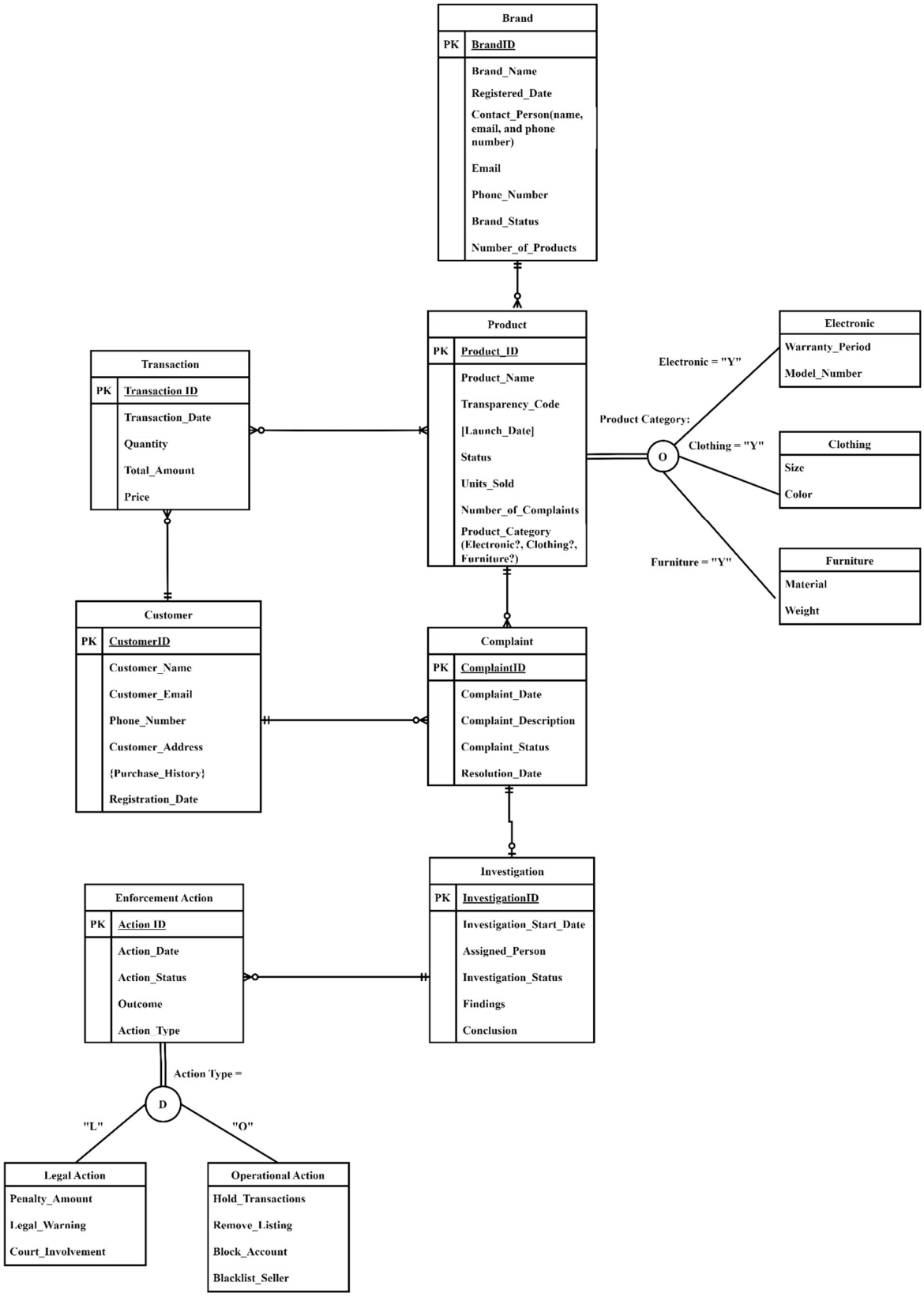
Attribute	**Data type**	**Constraints**
Hold_Transactions	Boolean	indicates if transactions are being held
Remove_Listing	Boolean	indicates if the product listing has been removed
Block_Account	Boolean	indicates if the customer or seller account is blocked
Blacklist_Seller	Boolean	indicates if the seller is blacklisted.

BUSINESS RULES

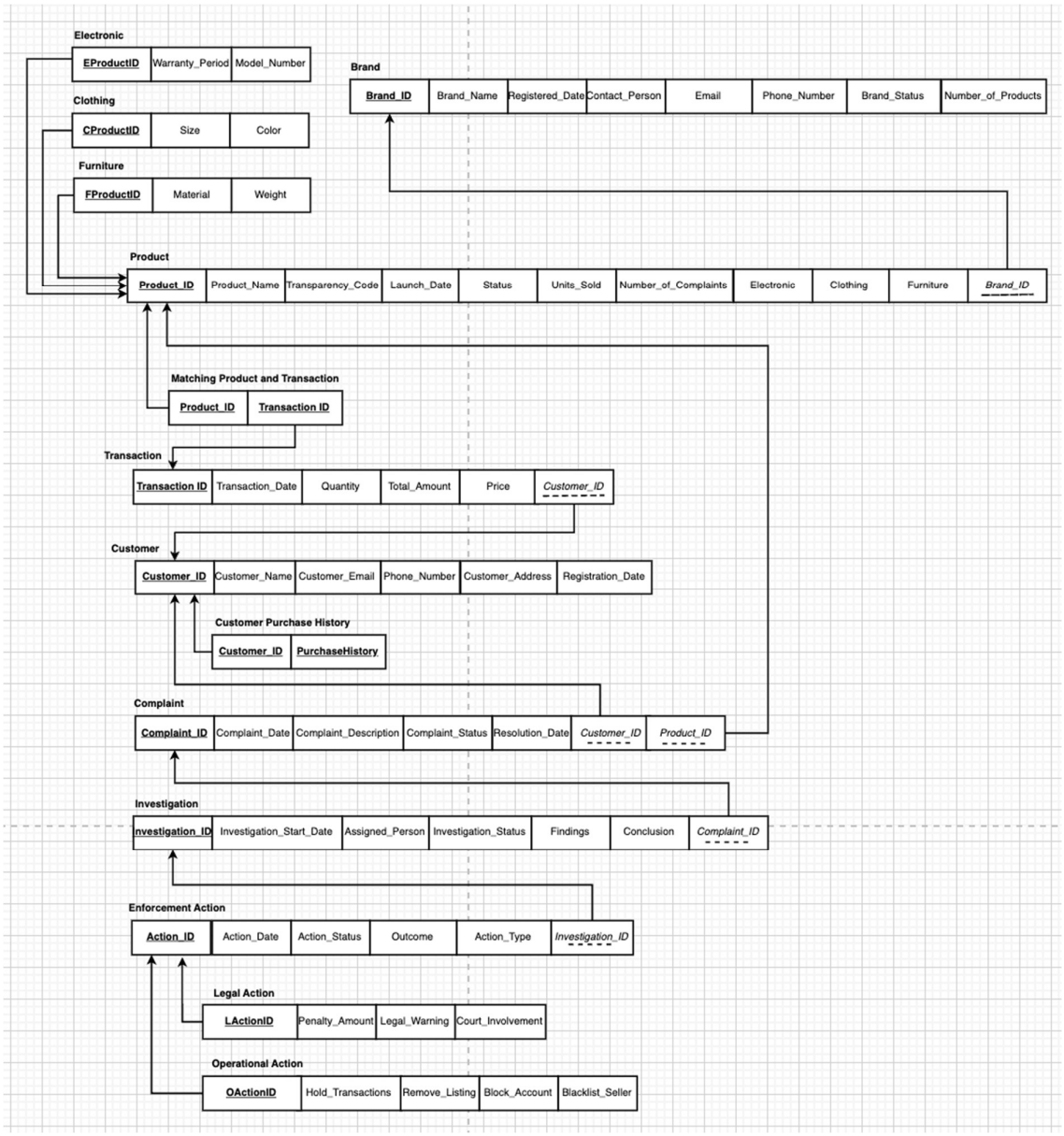
- Each brand can register many products
- Each product must be registered by only one brand.
- Each customer can file many complaints
- Each complaint must be filed by one customer.
- Each product may receive many complaints
- Each complaint must link to one product.
- Each complaint can lead to one investigation.
- Each investigation must be tied to precisely one complaint.
- Each investigation can result in many enforcement actions
- Each enforcement action must be related to exactly one investigation.
- Each customer can engage in many transactions
- Each transaction must involve exactly one customer.
- Each transaction must involve at least one product
- Each product can be involved in many transactions.

EER DIAGRAM & 3NF

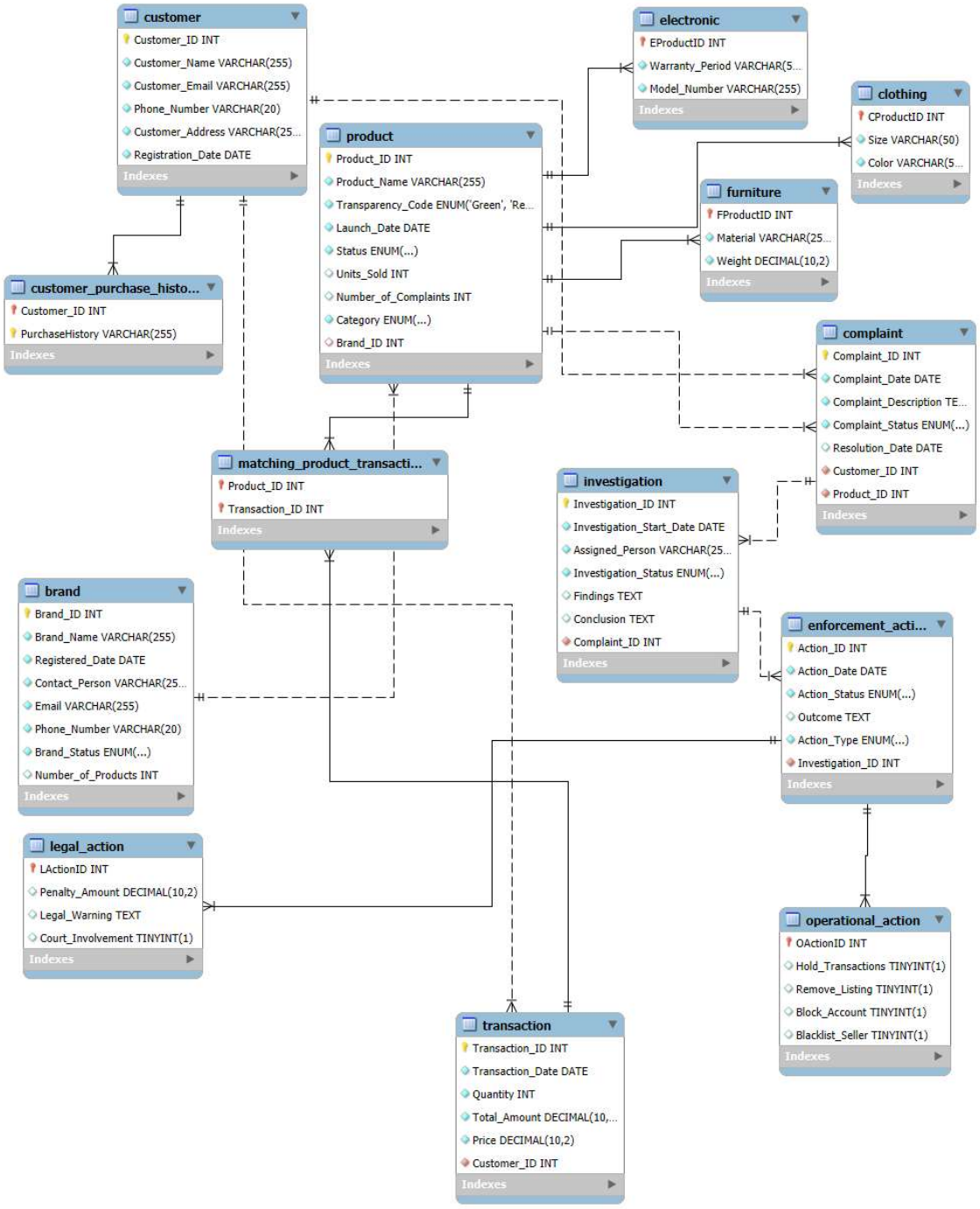
The EERD is provided in the following page – (All attributes are in bold because of the clarity of the image. Primary keys are mentioned Underlined)


Brand
PK
BrandID
Brand_Name
Registered_Date
Contact_Person(name, email, and phone number)
Email
Phone_Number
Brand_Status
Number_of_Products
Product
PK
Product_ID
Product_Name
Transparency_Code
[Launch_Date]
Status
Units_Sold
Number_of_Complaints
Product_Category (Electronic?, Clothing?, Furniture?)
Product Category:
Electronic = "Y"
Clothing = "Y"
Furniture = "Y"
O
Electronic
Warranty_Period
Model_Number
Clothing
Size
Color
Furniture
Material
Weight
Transaction
PK
Transaction ID
Transaction_Date
Quantity
Total_Amount
Price
Customer
PK
CustomerID
Customer_Name
Customer_Email
Phone_Number
Customer_Address
{Purchase_History}
Registration_Date
Complaint
PK
ComplaintID
Complaint_Date
Complaint_Description
Complaint_Status
Resolution_Date
Investigation
PK
InvestigationID
Investigation_Start_Date
Assigned_Person
Investigation_Status
Findings
Conclusion
Enforcement Action
PK
Action ID
Action_Date
Action_Status
Outcome
Action_Type
Action Type =
D
"L"
"O"
Legal Action
Penalty_Amount
Legal_Warning
Court_Involvement
Operational Action
Hold_Transactions
Remove_Listing
Block_Account
Blacklist_Seller

A relation model (normalised 3 NF) based on EERD.



A second ER Diagram using the Model feature of MySQL Workbench



INSIGHTS

---

1. Are there specific product categories more prone to fraud?

Purpose:

Identify which categories have the highest complaint rates to detect potential fraud-prone segments.

Logic:

- Sum up complaints and units sold per category.
- Calculate complaint rate as (complaints/units sold) * 100.
- Sort categories by complaint rate.

Insight:

Categories with high complaint rates (e.g., electronics, clothing) may have quality issues or fraudulent sellers.

```
347     -- 1. Are there specific product categories more prone to fraud?
348 •   SELECT
349         p.Category,
350         SUM(p.Units_Sold) AS Total_Units_Sold,
351         SUM(p.Number_of_Complaints) AS Total_Complaints,
352         ROUND((SUM(p.Number_of_Complaints) / SUM(p.Units_Sold)) * 100, 2) AS Complaint_Rate
353     FROM
354         Product p
355     GROUP BY
356         p.Category
357     ORDER BY
358         Complaint_Rate DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Category	Total_Units_Sold	Total_Complaints	Complaint_Rate
Electronic	5311862	672124	12.65
Clothing	5336078	98970	1.85
Furniture	3360388	3358	0.10

2. Are there any seasonal trends in complaint rates?

Purpose:

Understand if complaints vary by month to spot seasonal fraud patterns.

Logic:

- Count complaints per month.
- Calculate the percentage of complaints for each month relative to the total.
- Rank months by complaint volume.

Insight:

Seasonal spikes may indicate fraudulent activity during high-demand periods, e.g., holidays.

```sql
-- 2. Are there any seasonal trends in complaint rates?
SELECT
    MONTH(c.Complaint_Date) AS Complaint_Month,
    COUNT(c.Complaint_ID) AS Total_Complaints,
    ROUND(COUNT(c.Complaint_ID) * 100.0 / (SELECT COUNT(*) FROM Complaint), 2) AS Complaint_Percentage
FROM
    Complaint c
GROUP BY
    Complaint_Month
```

100% | 10:357 | 2 errors found

Result Grid | Filter Rows: Search | Export:

Complaint_Month	Total_Complaints	Complaint_Percentage
4	2	20.00
5	1	10.00
7	1	10.00
8	2	20.00
9	4	40.00

3. Which brands in these categories contribute to the high number of complaints?

Purpose:

Focus on brands responsible for the most complaints in fraud-prone categories.

Logic:

- Filter by categories identified in Question 1.
- Group by brand and category to measure each brand's complaint rate.

Insight:

Brands with the highest complaint rates are likely selling fraudulent products or have quality issues.

```sql
-- 3. Which brands in these categories contribute to the high number of complaints?
SELECT
    b.Brand_Name,
    p.Category,
    SUM(p.Units_Sold) AS Total_Units_Sold,
    SUM(p.Number_of_Complaints) AS Total_Complaints,
    ROUND((SUM(p.Number_of_Complaints) / SUM(p.Units_Sold)) * 100, 2) AS Complaint_Rate
FROM
    Brand b
JOIN
    Product p ON b.Brand_ID = p.Brand_ID
WHERE
    p.Category IN ('Electronic', 'Clothing')
GROUP BY
    b.Brand_Name, p.Category
ORDER BY
    Category,
    Complaint_Rate DESC;
```

Brand_Name	Category	Total_Units_Sold	Total_Complaints	Complaint_Rate
Real Amazon	Electronic	1892394	669000	35.35
Apple Store	Electronic	1593589	1592	0.10
Samsung	Electronic	941468	941	0.10
Amazon	Electronic	884411	591	0.07
Lulu Lemonade	Clothing	1363571	95000	6.97
Zara	Clothing	951980	951	0.10
Levi	Clothing	820671	820	0.10
Addidas	Clothing	1531477	1531	0.10
Abercrombie	Clothing	668379	668	0.10

4. Who is the contact person for these problematic brands?

Purpose:

Identify and profile the sellers of potentially fraudulent products.

Logic:

Directly fetch contact details for specific brands identified in Question 3.

Insight:

Having contact information aids in follow-up investigations and enforcement.

```
390     -- 4. Who is the contact person for these problematic brands?
391 •   SELECT
392         b.Brand_Name,
393         b.Contact_Person,
394         b.Email,
395         b.Phone_Number
396     FROM
397         Brand b
398     WHERE
399         b.Brand_Name IN ('Real Amazon', 'Lulu Lemonade');
400
401
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Brand_Name	Contact_Person	Email	Phone_Number
Real Amazon	Jeffy Bezos	Realjeff1994@yahoo.com	(724) 312-4018
Lulu Lemonade	Taylor Swiffy	Cruelwinter@gmail.com	(412) 293-2916

5. What are the specific customer complaints about these brands?

Purpose:

Understand the nature of complaints to gauge fraud severity and impact.

Logic:

- Join complaints with product and brand data.
- Filter by problematic brands.

Insight:

Complaint details provide actionable insights into specific fraudulent or defective products.

```sql
-- 5. What are the specific customer complaints about these brands?
SELECT
    c.Complaint_ID,
    c.Complaint_Description,
    c.Complaint_Status,
    c.Resolution_Date,
    p.Product_Name,
    b.Brand_Name
FROM
    Complaint c
JOIN
    Product p ON c.Product_ID = p.Product_ID
JOIN
    Brand b ON p.Brand_ID = b.Brand_ID
WHERE
    b.Brand_Name IN ('Real Amazon', 'Lulu Lemonade');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Complaint_ID	Complaint_Description	Complaint_Status	Resolution_Date	Product_Name	Brand_Name
1	Alexa cannot connect to wifi	Closed	2024-05-13	Alexa pro plus	Real Amazon
2	I call Alexa name but it does not respond to my ...	Closed	2024-05-15	Alexa 16 plus	Real Amazon
3	The Alexa is broken after I bought it only a wee...	Closed	2024-06-04	Alexa with Siri	Real Amazon
4	This is not a product from Lululemon. The finishi...	Rejected	2024-08-16	Tayloy Swift T-shirt	Lulu Lemonade
5	They are orange, not red!	In Progress	NULL	Red sock	Lulu Lemonade
6	They are not red socks	In Progress	NULL	Red sock	Lulu Lemonade

6. Have we taken any action to investigate these brands?

Purpose:

Assess the investigation status, findings, and results for the identified brands.

Logic:

- Trace complaints to investigations.
- Check actions, findings, and current status.

Insight:

Shows the efficiency and effectiveness of the organisation’s response to fraud.

```sql
-- 6. Have we taken any action to investigate these brands?
SELECT
    i.Investigation_ID,
    i.Assigned_Person,
    i.Investigation_Status,
    i.Findings,
    i.Conclusion,
    b.Brand_Name
FROM
    Investigation i
JOIN
    Complaint c ON i.Complaint_ID = c.Complaint_ID
JOIN
    Product p ON c.Product_ID = p.Product_ID
JOIN
    Brand b ON p.Brand_ID = b.Brand_ID
WHERE
    b.Brand_Name IN ('Real Amazon', 'Lulu Lemonade');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Investigation_ID	Assigned_Person	Investigation_Status	Findings	Conclusion	Brand_Name
1	Mitchael Keeney	Complete	Product imported with falsified documents	Legal action initiated	Real Amazon
2	Joscelin Cobleigh	Complete	Unauthorized seller	Seller account suspended	Real Amazon
3	Mindy Greenrde	Complete	Unauthorized seller	Seller account suspended	Real Amazon
4	Phyllys Cridlin	Complete	Not listed as a Lululemon seller	No action required	Lulu Lemonade
5	Sherri Vurley	Open	NULL	NULL	Lulu Lemonade
6	Jakob Bruffell	Open	NULL	NULL	Lulu Lemonade

7. What is the average time taken to investigate and close complaints?

Purpose:

Evaluate operational efficiency in resolving complaints.

Logic:

- Calculate the average days between complaint and resolution.
- Calculate the average days between the complaint and the investigation start.

Insight:

Helps track and improve the complaint-handling timeline, ensuring better customer satisfaction.

```
437     -- 7. What is the average time taken to investigate and close complaints?
438 •   SELECT
439         ROUND(AVG(DATEDIFF(c.Resolution_Date, c.Complaint_Date)), 2) AS Avg_Resolution_Time,
440         ROUND(AVG(DATEDIFF(i.Investigation_Start_Date, c.Complaint_Date)), 2) AS Avg_Investigation_Start_Time
441     FROM
442         Complaint c
443     JOIN
444         Investigation i ON c.Complaint_ID = i.Complaint_ID
445     WHERE
446         c.Complaint_Status = 'Closed';
447
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Avg_Resolution_Time	Avg_Investigation_Start_Time
19.80	2.00

COUNTERFEIT ASSESSMENT

"How effective is the current complaint resolution and investigation process in mitigating high-risk products and protecting the brand's reputation?"

Logic:

1. High-Risk Product Percentage:
    - Calculate the percentage of products flagged as high-risk (products with a complaint rate > 5% and more than 50 complaints). A high value indicates a significant concern about product quality or fraud risk.
2. Resolution Efficiency Percentage:
    - Assess the proportion of complaints that were successfully resolved (Complaint_Status = 'Closed'). Low efficiency signals room for improvement in customer service and problem resolution.
3. Average Resolution Time:
    - Calculate the average time (in days) taken to close complaints. A high average time indicates delays that could affect customer satisfaction.
4. Investigation Efficiency Percentage:
    - Evaluate how many investigations (Investigation_Status = 'Complete') were successfully closed as a percentage of all investigations. This is crucial for understanding whether fraud or quality concerns are promptly addressed.
5. Top Complaint Brand:
    - Identify the brand with the highest complaints, highlighting areas needing immediate focus.

Purpose:

This analysis serves to:

- Identify inefficiencies in the complaint resolution and investigation processes.
- Highlight high-risk products and their impact on customer trust and brand reputation.
- Direct, immediate actions toward the top offending brand to mitigate potential damage.
- Provide actionable insights to improve business practices, such as optimising resolution timelines and focusing on problematic product categories or brands.

Insights from the Results:

1. High-Risk Product Percentage (11.76%):
    - Indicates that some products are significantly problematic, as the percentage exceeds the number of distinct products. This suggests multiple overlapping issues (e.g., several high-risk complaints linked to the same product).
2. Resolution Efficiency Percentage (50%):
    - shows that a half of the complaints have been resolved, signalling a critical failure in the complaint-handling process. This is a significant area for improvement.
3. Average Resolution Time (19 days):
    - Indicates significant delays in resolving customer complaints. Reducing this time can improve customer satisfaction.
4. Investigation Efficiency Percentage (70%):
    - Indicates that approximately a two-third of investigations are being completed. Improving investigation processes is essential to address fraud and quality issues more effectively.
5. Top Complaint Brand (Lulu Lemonade):
    - This brand has the highest number of complaints, and addressing its issues should be the priority to protect the company's reputation and ensure customer trust.

Final Statement:

Current analysis reveals that while 11.76% of products are flagged as high-risk, 50% of the customer complaints have been resolved, and 70% of investigations are completed. The average resolution time of 19 days is significantly high, indicating inefficiencies in handling customer concerns. Immediate focus on Lulu Lemonade, the brand with the highest complaints, is essential to rebuild trust, improve operational processes, and protect the brand's reputation. A targeted approach to resolving complaints, expediting investigations, and addressing high-risk products is imperative to enhancing customer satisfaction and maintaining business integrity.

```sql
-- 1.High Risk Product Percentage
SELECT
    ROUND((SUM(CASE WHEN (p.Number_of_Complaints / p.Units_Sold) > 0.05 AND
    p.Number_of_Complaints > 50 THEN 1 ELSE 0 END)/ COUNT(DISTINCT p.Product_ID))* 100, 2)
    AS High_Risk_Product_Percentage
FROM Product p
LEFT JOIN Brand b ON p.Brand_ID = b.Brand_ID
WHERE b.Brand_Status = 'active';

```

100% 1:449 2 errors found

Result Grid Filter Rows: Search Export:

High_Risk_Product_Percent...
11.76

```sql
-- 2.Resolution efficiency percentage
SELECT
    ROUND((SUM(CASE WHEN c.Complaint_Status = 'Closed' THEN 1 ELSE 0 END)
    / COUNT(c.Complaint_ID)) * 100, 2) AS Resolution_Efficiency_Percentage
FROM Complaint c;

```

100% 1:460 2 errors found

Result Grid Filter Rows: Search Export:

Resolution_Efficiency_Percentage
50.00

```sql
-- 3. Average resolution time
SELECT
    ROUND(AVG(DATEDIFF(c.Resolution_Date, c.Complaint_Date)), 2) AS Avg_Resolution_Time
FROM Complaint c
WHERE c.Complaint_Status = 'Closed';

```

100% 1:467 2 errors found

Result Grid Filter Rows: Search Export:

Avg_Resolution_Time
19.80

```sql
-- 4.Investigate efficiency percentage
SELECT
    ROUND((SUM(CASE WHEN i.Investigation_Status = 'Complete' THEN 1 ELSE 0 END)
    / COUNT(i.Investigation_ID)) * 100, 2) AS Investigation_Efficiency_Percentage
FROM Investigation i;

```

100% 37:472 2 errors found

Result Grid Filter Rows: Search Export:

Investigation_Efficiency_Percent...
70.00

```sql
-- 5.Top complaint brand (Active)
SELECT
    b.Brand_Name AS Top_Complaint_Brand,
    COUNT(c.Complaint_ID) AS Total_Complaints
FROM Brand b
JOIN Product p ON p.Brand_ID = b.Brand_ID
JOIN Complaint c ON c.Product_ID = p.Product_ID
WHERE
    b.Brand_Status = 'active'
GROUP BY
    b.Brand_Name
ORDER BY
    Total_Complaints DESC
LIMIT 1;
```

100% | 1:479 | 2 errors found

Result Grid | Filter Rows: Search | Export: | Fetch rows:

Top_Complaint_Brand	Total_Complaints
Lulu Lemonade	3

CONCLUSION

This project provided significant benefits and posed valuable challenges that enhanced our learning experience. By designing and implementing a database for Amazon's Brand Protection initiative, we gained a deep understanding of database management, particularly in entity-relationship modelling, normalisation, and SQL query optimisation. The project also enhanced our knowledge about organisational structures, specifically how data flows within a large-scale initiative like Amazon's Transparency Program. By simulating real-world scenarios, we developed practical insights into the lifecycle of product authenticity, from brand registration to enforcement actions.

However, the project was not without its challenges. One of the primary difficulties was ensuring that the generated data aligned with the project's conceptual requirements while maintaining logical consistency across 14 interconnected tables. To address this, we iteratively refined the data, using tools like Mockaroo and Excel to validate relationships and ensure proper table linkage. Another challenge was creating SQL scripts that supported complex queries and calculations, such as complaint trends and enforcement outcomes. This required extensive testing and debugging to ensure accuracy and reliability.

These challenges strengthened our problem-solving skills and deepened our understanding of database systems. By overcoming these obstacles, we delivered a robust and functional

database that supports the objectives of counterfeit detection and brand protection while gaining invaluable hands-on experience for future projects.

RECOMMENDATIONS

---

1. Add a Categorizing Attribute to the Complaint Table

   Rationale: Categorizing the reasons for complaints (e.g., Defective Product, Counterfeit, Late Delivery, Mismatched Description) will enable a more granular analysis of complaint trends. This can help identify specific areas for improvement in product quality, logistics, or seller behaviour.

   Value:

   - ➢ Identify recurring complaint types for targeted interventions (e.g., if Counterfeit is frequent, prioritise seller authentication processes).
   - ➢ Help focus investigations on the most impactful issues (e.g., prioritise complaints about Counterfeit over Late Delivery).

2. Enhance Customer Data

   Rationale: Adding demographic attributes like age, gender, income level, and location to the Customer table will allow for a deeper analysis of customer behaviour and complaint patterns. This will help tailor solutions and allocate resources more effectively.

   Value:

   a. **Age Analysis:**
      - ❖ Are younger customers more likely to complain about electronics than older customers?
      - ❖ What percentage of complaints from specific age groups are valid or resolved?

   b. **Gender Insights:**
      - ❖ Do gender differences exist in complaint behaviour (e.g., are women more likely to complain about clothing and men about electronics)?

   c. **Geographic Trends:**
      - ❖ Do complaints cluster in specific regions or cities? For example, is fraud more prevalent in urban versus rural areas?

APPENDIX

SQL Code

Creating Tables

```
CREATE SCHEMA AmazonBrands_db;
USE AmazonBrands_db;

CREATE TABLE Brand (
    Brand_ID INT PRIMARY KEY,
    Brand_Name VARCHAR(255) NOT NULL,
    Registered_Date DATE NOT NULL,
    Contact_Person VARCHAR(255) NOT NULL,
    Email VARCHAR(255) NOT NULL UNIQUE,
    Phone_Number VARCHAR(20) NOT NULL,
    Brand_Status ENUM('active', 'suspended') NOT NULL,
    Number_of_Products INT DEFAULT 0
);

CREATE TABLE Product (
    Product_ID INT PRIMARY KEY,
    Product_Name VARCHAR(255) NOT NULL,
    Transparency_Code ENUM('Green', 'Red') NOT NULL,
    Launch_Date DATE NOT NULL,
    Status ENUM('Available', 'Unavailable') NOT NULL,
    Units_Sold INT DEFAULT 0,
    Number_of_Complaints INT DEFAULT 0,
    Category ENUM('Electronic', 'Clothing', 'Furniture') NOT NULL,
    Brand_ID INT,
    FOREIGN KEY (Brand_ID) REFERENCES Brand(Brand_ID)
);

CREATE TABLE Electronic (
```

```
    EProductID INT PRIMARY KEY,
    Warranty_Period VARCHAR(50) NOT NULL,
    Model_Number VARCHAR(255) NOT NULL,
    FOREIGN KEY (EProductID) REFERENCES Product(Product_ID)
);

CREATE TABLE Clothing (
    CProductID INT PRIMARY KEY,
    Size VARCHAR(50) NOT NULL,
    Color VARCHAR(50) NOT NULL,
    FOREIGN KEY (CProductID) REFERENCES Product(Product_ID)
);

CREATE TABLE Furniture (
    FProductID INT PRIMARY KEY,
    Material VARCHAR(255) NOT NULL,
    Weight DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (FProductID) REFERENCES Product(Product_ID)
);

CREATE TABLE Customer (
    Customer_ID INT PRIMARY KEY,
    Customer_Name VARCHAR(255) NOT NULL,
    Customer_Email VARCHAR(255) NOT NULL UNIQUE,
    Phone_Number VARCHAR(20) NOT NULL,
    Customer_Address VARCHAR(255) NOT NULL,
    Registration_Date DATE NOT NULL
);

CREATE TABLE Transaction (
    Transaction_ID INT AUTO_INCREMENT PRIMARY KEY,
    Transaction_Date DATE NOT NULL,
    Quantity INT NOT NULL,
```

```
    Total_Amount DECIMAL(10, 2) NOT NULL,
    Price DECIMAL(10, 2) NOT NULL,
    Customer_ID INT NOT NULL,
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)
);

CREATE TABLE Matching_Product_Transaction (
    Product_ID INT NOT NULL,
    Transaction_ID INT NOT NULL,
    PRIMARY KEY (Product_ID, Transaction_ID),
    FOREIGN KEY (Product_ID) REFERENCES Product(Product_ID),
    FOREIGN KEY (Transaction_ID) REFERENCES Transaction(Transaction_ID)
);

CREATE TABLE Customer_Purchase_History (
    Customer_ID INT NOT NULL,
    PurchaseHistory VARCHAR(255) NOT NULL,
    PRIMARY KEY (Customer_ID, PurchaseHistory),
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)
);

CREATE TABLE Complaint (
    Complaint_ID INT PRIMARY KEY,
    Complaint_Date DATE NOT NULL,
    Complaint_Description TEXT NOT NULL,
    Complaint_Status ENUM('Closed', 'Rejected', 'In Progress') NOT NULL,
    Resolution_Date DATE DEFAULT NULL,
    Customer_ID INT NOT NULL,
    Product_ID INT NOT NULL,
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID),
    FOREIGN KEY (Product_ID) REFERENCES Product(Product_ID)
);
```

```
CREATE TABLE Investigation (
    Investigation_ID INT PRIMARY KEY,
    Investigation_Start_Date DATE NOT NULL,
    Assigned_Person VARCHAR(255) NOT NULL,
    Investigation_Status ENUM('Open', 'In Progress', 'Complete') NOT NULL,
    Findings TEXT DEFAULT NULL,
    Conclusion TEXT DEFAULT NULL,
    Complaint_ID INT NOT NULL,
    FOREIGN KEY (Complaint_ID) REFERENCES Complaint(Complaint_ID)
);

CREATE TABLE Enforcement_Action (
    Action_ID INT PRIMARY KEY,
    Action_Date DATE NOT NULL,
    Action_Status ENUM('Approved', 'Rejected', 'On Hold') NOT NULL,
    Outcome TEXT DEFAULT NULL,
    Action_Type ENUM('Legal', 'Operational') NOT NULL,
    Investigation_ID INT NOT NULL,
    FOREIGN KEY (Investigation_ID) REFERENCES Investigation(Investigation_ID)
);
CREATE TABLE Legal_Action (
    LActionID INT PRIMARY KEY,
    Penalty_Amount DECIMAL(10, 2) DEFAULT NULL,
    Legal_Warning TEXT DEFAULT NULL,
    Court_Involvement BOOLEAN DEFAULT FALSE,
    FOREIGN KEY (LActionID) REFERENCES Enforcement_Action(Action_ID)
);

CREATE TABLE Operational_Action (
    OActionID INT PRIMARY KEY,
    Hold_Transactions BOOLEAN DEFAULT FALSE,
    Remove_Listing BOOLEAN DEFAULT FALSE,
    Block_Account BOOLEAN DEFAULT FALSE,
```

```
    Blacklist_Seller BOOLEAN DEFAULT FALSE,
    FOREIGN KEY (OActionID) REFERENCES Enforcement_Action(Action_ID)
);
```

Inserting Data

```
INSERT INTO Brand (Brand_ID, Brand_Name, Registered_Date, Contact_Person, Email,
Phone_Number, Brand_Status, Number_of_Products)
VALUES
(1, 'Apple Store', '2022-01-23', 'Benn Palke', 'Benn.p@apple.com', '(757) 911-7615', 'active',
2),
(2, 'Amazon', '2022-01-31', 'Claudell Philipeaux', 'Claudellphilip@amazon', '(661) 723-1854',
'active', 2),
(3, 'Real Amazon', '2022-02-18', 'Jeffy Bezos', 'Realjeff1994@yahoo.com', '(724) 312-4018',
'suspended', 3),
(4, 'Zara', '2022-04-29', 'Saundra Blaza', 'Saundra.b@zara.com', '(614) 128-4933', 'active', 1),
(5, 'Levi', '2022-06-14', 'Orelie Deye', 'Orelie.d@levi.com', '(254) 744-3498', 'active', 1),
(6, 'Lulu Lemonade', '2022-06-30', 'Taylor Swiffy', 'Cruelwinter@gmail.com', '(412) 293-
2916', 'active', 2),
(7, 'IKEA', '2022-07-28', 'Cissiee Buttel', 'Cissiee1980@ikea.com', '(702) 211-4848', 'active',
1),
(8, 'Lowe', '2022-07-29', 'Sebastien Farfolomeev', 'Sebastien.far@lowe.com', '(202) 776-
0642', 'active', 1),
(9, 'Pottery Barn', '2022-10-15', 'Mychal Leitch', 'Mychal.leitch@potterybarn.com', '(203)
276-8548', 'active', 1),
(10, 'Addidas', '2022-10-31', 'Ollie Sidnell', 'ollie.sidnell@addidas.com', '(915) 494-8354',
'active', 1),
(11, 'Home Living', '2022-12-23', 'Cleavland Frankiss', 'cleavland.frankiss@live.com', '(813)
460-8860', 'active', 1),
(12, 'Samsung', '2023-01-01', 'Horton Quarrington', 'horton.quarrington@samsung.com',
'(305) 035-3258', 'active', 1),
(13, 'Abercrombie', '2023-02-01', 'Anissa Chmarny', 'anissa.chmarny@abercrombie.com',
'(417) 168-0567', 'active', 1);
INSERT INTO Product (Product_ID, Brand_ID, Product_Name, Transparency_Code,
Category, Launch_Date, Status, Units_Sold, Number_of_Complaints)
```

```
VALUES
(1, 1, 'iPhone 16 charger', 'Green', 'Electronic', '2023-05-08', 'Available', 638902, 638),
(2, 1, 'Macbook pro', 'Green', 'Electronic', '2023-08-08', 'Available', 954687, 954),
(3, 2, 'Alexa', 'Green', 'Electronic', '2023-10-04', 'Available', 91802, 91),
(4, 2, 'Alexa 2.0', 'Green', 'Electronic', '2023-11-20', 'Available', 792609, 500),
(5, 3, 'Alexa pro plus', 'Red', 'Electronic', '2024-01-01', 'Available', 595603, 99000),
(6, 3, 'Alexa 16 plus', 'Red', 'Electronic', '2024-01-06', 'Available', 748602, 315000),
(7, 3, 'Alexa with Siri', 'Red', 'Electronic', '2024-01-11', 'Available', 548189, 255000),
(8, 4, 'Red Christmas sweat', 'Green', 'Clothing', '2024-01-27', 'Available', 951980, 951),
(9, 5, 'Blue jean', 'Green', 'Clothing', '2024-03-25', 'Available', 820671, 820),
(10, 6, 'Tayloy Swift T-shirt', 'Green', 'Clothing', '2024-04-04', 'Available', 797987, 45000),
(11, 6, 'Red sock', 'Green', 'Clothing', '2024-04-12', 'Available', 565584, 50000),
(12, 7, 'Dinning table', 'Green', 'Furniture', '2024-05-05', 'Available', 947860, 947),
(13, 8, 'Bed frame', 'Green', 'Furniture', '2024-06-03', 'Available', 882804, 882),
(14, 9, 'Shoe rack', 'Green', 'Furniture', '2024-06-06', 'Available', 979317, 979),
(15, 10, 'Stan Smith short', 'Green', 'Clothing', '2024-06-25', 'Available', 156017, 156),
(16, 10, 'Addidas Sportswear', 'Green', 'Clothing', '2024-06-29', 'Available', 491050, 491),
(17, 11, 'Benchwright Extending Dining Table', 'Green', 'Furniture', '2024-07-08', 'Available',
550407, 550),
(18, 10, 'Essential Sweatpant', 'Green', 'Clothing', '2024-07-12', 'Available', 884410, 884),
(19, 12, 'Galaxy Buds3 Pro', 'Green', 'Electronic', '2024-07-19', 'Available', 941468, 941),
(20, 13, 'Sherpa-Lined Zip Workwear Jacket', 'Green', 'Clothing', '2024-08-15', 'Available',
668379, 668);

INSERT INTO Electronic (EProductID, Warranty_Period, Model_Number)
VALUES
(1, 12, 'IX0437-E1'),
(2, 12, 'GU4981-G6'),
(3, 24, 'BG0663-T0'),
(4, 24, 'DP8553-B7'),
(5, 6, 'BP5935-M4'),
(6, 6, 'BD3907-O9'),
(7, 3, 'BE4265-D4'),
```

(19, 12, 'SSB2024-A2');

INSERT INTO Clothing (CProductID, Size, Color)
VALUES
(8, 'XXL', 'Red'),
(9, 'S', 'Blue'),
(10, 'M', 'Red'),
(11, 'M', 'Red'),
(15, 'L', 'Green'),
(16, 'L', 'Black'),
(18, 'L', 'Base'),
(20, 'S', 'Pink');

INSERT INTO Furniture (FProductID, Material, Weight)
VALUES
(12, 'Plastic', 214),
(13, 'Wood', 250),
(14, 'Bamboo', 745),
(17, 'Wood', 1000);

INSERT INTO Customer (Customer_ID, Customer_Name, Customer_Email,
Phone_Number, Customer_Address, Registration_Date)
VALUES
(1, 'Karmen Tizard', 'karmen.tizard@gmail.com', '(309) 565-6638', '5760 Grey Dove Lane,
Peoria, Illinois, United States, 61651', '2023-04-04'),
(2, 'Ky Rasher', 'ky.rasher@hotmail.fr', '(816) 632-7665', '7990 Black Stone Place, Kansas
City, Missouri, United States, 64130', '2023-05-29'),
(3, 'Raynor Meffin', 'raynor.meffin@gmail.com', '(254) 513-1733', '13683 Venable Court,
Waco, Texas, United States, 76705', '2023-10-15'),
(4, 'Judye Pebworth', 'judye.pebworth@yahoo.com', '(228) 970-7444', '1385 Peachtree
Avenue, Biloxi, Mississippi, United States, 39534', '2023-12-19'),
(5, 'Whittaker Press', 'whittaker.press@t-online.de', '(212) 774-6875', '8223 Dyals Court, New
York City, New York, United States, 10260', '2023-06-03'),

(6, 'Daveta OLoughlin', 'daveta.oloughlin@wanadoo.fr', '(310) 013-0875', '9959 Charlene Place, Palo Alto, California, United States, 94302', '2023-07-16'),
(7, 'Matthus DAlessio', 'matthus.dalessio@gmail.com', '(412) 253-3093', '8492 Donna Avenue, Pittsburgh, Pennsylvania, United States, 15220', '2023-06-10'),
(8, 'Delly Falvey', 'delly.falvey@yahoo.com', '(860) 309-7698', '1530 Thistledown Way, West Hartford, Connecticut, United States, 06127', '2023-12-23'),
(9, 'Eirena Benneton', 'eirena.benneton@gmail.com', '(304) 534-8410', '11399 Barnhill Lane, Huntington, West Virginia, United States, 25716', '2023-08-20'),
(10, 'Khalil Warrack', 'khalil.warrack@tiscali.it', '(484) 484-9924', '10997 Linden Lane, Reading, Pennsylvania, United States, 19605', '2023-08-10'),
(11, 'Archaimbaud Tointon', 'archaimbaud.tointon@hotmail.fr', '(937) 875-2039', '2522 Janeann Terrace, Dayton, Ohio, United States, 45403', '2024-09-23'),
(12, 'Tailor Yann', 'tailor.yann@yahoo.com', '(952) 949-6006', '9687 Huckleberry Street, Young America, Minnesota, United States, 55573', '2024-10-07'),
(13, 'Ban Blanchette', 'ban.blanchette@yahoo.com', '(831) 867-0228', '10710 Oxbow Drive, Santa Cruz, California, United States, 95064', '2024-07-02'),
(14, 'Lyda Cardon', 'lyda.cardon@gmail.com', '(775) 724-2859', '13248 Dillon Lane, Reno, Nevada, United States, 89505', '2024-07-30'),
(15, 'Lindsey Whytock', 'lindsey.whytock@hotmail.com', '(818) 467-9430', '9953 Varnville Way, Glendale, California, United States, 91205', '2024-05-26');

INSERT INTO Transaction (Transaction_ID, Customer_ID, Transaction_Date, Quantity, Price, Total_Amount)
VALUES
(1, 1, '2024-01-08', 3, 50, 150),
(2, 1, '2024-02-08', 1, 1500, 1500),
(3, 2, '2024-02-11', 5, 80, 400),
(4, 3, '2024-03-11', 2, 100, 200),
(5, 4, '2024-04-11', 5, 15, 75),
(6, 5, '2024-04-11', 3, 20, 60),
(7, 6, '2024-05-15', 4, 30, 120),
(8, 7, '2024-06-11', 1, 150, 150),
(9, 7, '2024-06-19', 1, 129, 129),

```
(10, 8, '2024-06-29', 1, 200, 200),
(11, 9, '2024-07-29', 4, 5, 20),
(12, 10, '2024-08-20', 1, 550, 550),
(13, 10, '2024-08-23', 1, 650, 650),
(14, 10, '2024-08-22', 2, 150, 300),
(15, 10, '2024-08-24', 2, 5, 10),
(16, 11, '2024-08-25', 1, 229, 229),
(17, 12, '2024-08-26', 2, 115, 230),
(18, 13, '2024-08-27', 1, 1499, 1499),
(19, 12, '2024-08-18', 1, 99, 99),
(20, 14, '2024-08-29', 1, 199, 199),
(21, 15, '2024-08-30', 1, 169, 169);

INSERT INTO Matching_Product_Transaction (Product_ID, Transaction_ID)
VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5),
(6, 6),
(7, 7),
(8, 8),
(9, 9),
(10, 10),
(11, 11),
(12, 12),
(13, 13),
(14, 14),
(11, 15),
(15, 16),
(16, 17),
(17, 18),
```

```
(18, 19),
(19, 20),
(20, 21);

INSERT INTO Customer_Purchase_History (Customer_ID, PurchaseHistory)
VALUES
(1, 'iPhone 16 charger'),
(1, 'Macbook pro'),
(2, 'Alexa'),
(3, 'Alexa 2.0'),
(4, 'Alexa pro plus'),
(5, 'Alexa 16 plus'),
(6, 'Alexa with Siri'),
(7, 'Red Christmas sweat'),
(7, 'Blue jean'),
(8, 'Tayloy Swift T-shirt'),
(9, 'Red sock'),
(10, 'Dinning table'),
(10, 'Bed frame'),
(10, 'Shoe rack'),
(10, 'Red sock'),
(11, 'Stan Smith short'),
(12, 'Addidas Sportswear'),
(13, 'Benchwright Extending Dining Table'),
(12, 'Essential Sweatpant'),
(14, 'Galaxy Buds3 Pro'),
(15, 'Sherpa-Lined Zip Workwear Jacket');

INSERT INTO Complaint (Complaint_ID, Customer_ID, Product_ID, Complaint_Date,
Complaint_Description, Complaint_Status, Resolution_Date)
VALUES
(1, 4, 5, '2024-04-13', 'Alexa cannot connect to wifi', 'Closed', '2024-05-13'),
```

(2, 5, 6, '2024-04-15', 'I call Alexa name but it does not respond to my command', 'Closed', '2024-05-15'),
(3, 6, 7, '2024-05-22', 'The Alexa is broken after I bought it only a week ago', 'Closed', '2024-06-04'),
(4, 8, 10, '2024-07-03', 'This is not a product from Lululemon. The finishing is very bad', 'Rejected', '2024-08-16'),
(5, 9, 11, '2024-08-03', 'They are orange, not red!', 'In Progress', NULL),
(6, 10, 11, '2024-08-25', 'They are not red socks', 'In Progress', NULL),
(7, 11, 15, '2024-09-23', 'The green color of the short is not what I expected', 'Rejected', '2024-09-25'),
(8, 13, 17, '2024-09-24', 'The table height is much higher than the picture on Amazon', 'Closed', '2024-10-01'),
(9, 14, 19, '2024-09-26', 'The battery of earbuds is running out fast. It does not last 8 hours as marketed.', 'Closed', '2024-10-15'),
(10, 15, 20, '2024-09-27', 'The jacket is slightly bigger.', 'In Progress', NULL);

INSERT INTO Investigation (Investigation_ID, Investigation_Start_Date, Assigned_Person, Investigation_Status, Findings, Conclusion, Complaint_ID)
VALUES
(1, '2024-04-16', 'Mitchael Keeney', 'Complete', 'Product imported with falsified documents', 'Legal action initiated', 1),
(2, '2024-04-18', 'Joscelin Cobleigh', 'Complete', 'Unauthorized seller', 'Seller account suspended', 2),
(3, '2024-05-26', 'Mindy Greenrde', 'Complete', 'Unauthorized seller', 'Seller account suspended', 3),
(4, '2024-08-15', 'Phyllys Cridlin', 'Complete', 'Not listed as a Lululemon seller', 'No action required', 4),
(5, '2024-08-25', 'Sherri Vurley', 'Open', NULL, NULL, 5),
(6, '2024-08-25', 'Jakob Bruffell', 'Open', NULL, NULL, 6),
(7, '2024-09-23', 'Gaby Alexis', 'Complete', 'Color matches product description', NULL, 7),
(8, '2024-09-24', 'Tally Tapenden', 'Complete', 'Table height off by 5 cm', NULL, 8),
(9, '2024-09-26', 'Alexandrina Spurden', 'Complete', 'Battery drains faster than expected', NULL, 9),

(10, '2024-09-27', 'Brnaby Terbeek', 'Open', NULL, NULL, 10);

INSERT INTO Enforcement_Action (Action_ID, Action_Date, Action_Status, Outcome, Action_Type, Investigation_ID)
VALUES
(1, '2024-05-13', 'Approved', 'Legal actions executed', 'Legal', 1),
(2, '2024-05-15', 'Approved', 'Seller account suspended', 'Legal', 2),
(3, '2024-06-04', 'Approved', 'Seller account suspended', 'Legal', 3),
(4, '2024-09-15', 'Rejected', 'Seller intent not fraudulent', 'Operational', 4),
(5, '2024-09-25', 'On Hold', NULL, 'Operational', 5),
(6, '2024-09-25', 'On Hold', NULL, 'Operational', 6),
(7, '2024-09-25', 'Rejected', 'Seller intent not fraudulent', 'Operational', 7),
(8, '2024-10-01', 'Approved', 'Hold transaction until quality improves', 'Operational', 8),
(9, '2024-10-15', 'Approved', 'Hold transaction until quality improves', 'Operational', 9);

INSERT INTO Legal_Action (LActionID, Penalty_Amount, Legal_Warning, Court_Involvement)
VALUES
(1, 100000, 'Falsified documentation.', 1),
(2, 75000, '1st degree false branding.', 0),
(3, 50000, '2nd degree false branding.', 0);

INSERT INTO Operational_Action (OActionID, Hold_Transactions, Remove_Listing, Block_Account, Blacklist_Seller)
VALUES
(4, 0, 0, 0, 0),
(5, 1, 0, 0, 0),
(6, 1, 0, 0, 0),
(7, 0, 0, 0, 0),
(8, 1, 0, 1, 1),
(9, 1, 0, 1, 1);

Insight 1:

```
SELECT
    p.Category,
    SUM(p.Units_Sold) AS Total_Units_Sold,
    SUM(p.Number_of_Complaints) AS Total_Complaints,
    ROUND((SUM(p.Number_of_Complaints) / SUM(p.Units_Sold)) * 100, 2) AS
Complaint_Rate
FROM
    Product p
GROUP BY
    p.Category
ORDER BY
    Complaint_Rate DESC;
```

Insight 2:

```
SELECT
    MONTH(c.Complaint_Date) AS Complaint_Month,
    COUNT(c.Complaint_ID) AS Total_Complaints,
    ROUND(COUNT(c.Complaint_ID) * 100.0 / (SELECT COUNT(*) FROM Complaint), 2)
AS Complaint_Percentage
FROM
    Complaint c
GROUP BY
    Complaint_Month
```

Insight 3:

```
SELECT
    b.Brand_Name,
    p.Category,
```

```
    SUM(p.Units_Sold) AS Total_Units_Sold,
    SUM(p.Number_of_Complaints) AS Total_Complaints,
    ROUND((SUM(p.Number_of_Complaints) / SUM(p.Units_Sold)) * 100, 2) AS
Complaint_Rate
FROM
    Brand b
JOIN
    Product p ON b.Brand_ID = p.Brand_ID
WHERE
    p.Category IN ('Electronic', 'Clothing')
GROUP BY
    b.Brand_Name, p.Category
ORDER BY
    Category,
    Complaint_Rate DESC;
```

Insight 4:

```
SELECT
    b.Brand_Name,
    b.Contact_Person,
    b.Email,
    b.Phone_Number
FROM
    Brand b
WHERE
    b.Brand_Name IN ('Real Amazon', 'Lulu Lemonade');
```

Insight 5:

```
SELECT
    c.Complaint_ID,
    c.Complaint_Description,
```

```
    c.Complaint_Status,
    c.Resolution_Date,
    p.Product_Name,
    b.Brand_Name
FROM
    Complaint c
JOIN
    Product p ON c.Product_ID = p.Product_ID
JOIN
    Brand b ON p.Brand_ID = b.Brand_ID
WHERE
    b.Brand_Name IN ('Real Amazon', 'Lulu Lemonade');
```

Insight 6:

```
SELECT
    i.Investigation_ID,
    i.Assigned_Person,
    i.Investigation_Status,
    i.Findings,
    i.Conclusion,
    b.Brand_Name
FROM
    Investigation i
JOIN
    Complaint c ON i.Complaint_ID = c.Complaint_ID
JOIN
    Product p ON c.Product_ID = p.Product_ID
JOIN
    Brand b ON p.Brand_ID = b.Brand_ID
WHERE
    b.Brand_Name IN ('Real Amazon', 'Lulu Lemonade');
```

Insight 7:

```
SELECT
   ROUND(AVG(DATEDIFF(c.Resolution_Date, c.Complaint_Date)), 2) AS
Avg_Resolution_Time,
   ROUND(AVG(DATEDIFF(i.Investigation_Start_Date, c.Complaint_Date)), 2) AS
Avg_Investigation_Start_Time
FROM
   Complaint c
JOIN
   Investigation i ON c.Complaint_ID = i.Complaint_ID
WHERE
   c.Complaint_Status = 'Closed';
```

Counterfeit Assessment Matrices

1.High Risk Product Percentage

```
SELECT

   ROUND((SUM(CASE WHEN (p.Number_of_Complaints / p.Units_Sold) > 0.05 AND
p.Number_of_Complaints > 50 THEN 1 ELSE 0 END)/ COUNT(DISTINCT
p.Product_ID))* 100, 2) AS High_Risk_Product_Percentage

FROM Product p

LEFT JOIN Brand b ON p.Brand_ID = b.Brand_ID

WHERE b.Brand_Status = 'active';
```

2.Resolution efficiency percentage

```
SELECT

   ROUND((SUM(CASE WHEN c.Complaint_Status = 'Closed' THEN 1 ELSE 0 END) /
COUNT(c.Complaint_ID)) * 100, 2) AS Resolution_Efficiency_Percentage

FROM Complaint c;
```

3. Average resolution time

```
SELECT
   ROUND(AVG(DATEDIFF(c.Resolution_Date, c.Complaint_Date)), 2) AS
Avg_Resolution_Time
FROM Complaint c
WHERE c.Complaint_Status = 'Closed';
```

4.Investigate efficiency percentage

```
SELECT
   ROUND((SUM(CASE WHEN i.Investigation_Status = 'Complete' THEN 1 ELSE 0 END)
   / COUNT(i.Investigation_ID)) * 100, 2) AS Investigation_Efficiency_Percentage
FROM Investigation i;
```

5.Top complaint brand (Active)

```
SELECT
   b.Brand_Name AS Top_Complaint_Brand,
   COUNT(c.Complaint_ID) AS Total_Complaints
FROM Brand b
JOIN Product p ON p.Brand_ID = b.Brand_ID
JOIN Complaint c ON c.Product_ID = p.Product_ID
WHERE
   b.Brand_Status = 'active'
GROUP BY
```

```
    b.Brand_Name
ORDER BY
    Total_Complaints DESC
LIMIT 1;
```