Use Case-1: Building a Cart Analysis for Myph

1. Use Case Overview
Myph has launched a new phone range. Each product belongs to a category. Customers add products to a shopping cart. We aim to: Build a data model that supports products, categories, and carts. Allow category-based filtering. Support cart analytics, including outlier detection (surplus selections). Ensure transaction recovery and reliability.

2. Database Schema Design
Below are the SQL statements to create the relational data model.

Categories Table

```
CREATE TABLE Categories (
    category_id INT PRIMARY KEY AUTO_INCREMENT,
    category_name VARCHAR(100) NOT NULL,
    parent_category_id INT NULL,
    category_path VARCHAR(255),
    FOREIGN KEY (parent_category_id) REFERENCES Categories(category_id)
);
```

Products Table

```
CREATE TABLE Products (
    product_id INT PRIMARY KEY AUTO_INCREMENT,
    category_id INT NOT NULL,
    product_name VARCHAR(150) NOT NULL,
    description TEXT,
    stock_quantity INT DEFAULT 0,
    price DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (category_id) REFERENCES Categories(category_id)
);
```

Customers Table

```
CREATE TABLE Customers (
    customer_id INT PRIMARY KEY AUTO_INCREMENT,
    customer_name VARCHAR(150),
    email VARCHAR(150) UNIQUE
);
```

Cart Table

```
CREATE TABLE Cart (
    cart_id INT PRIMARY KEY AUTO_INCREMENT,
    customer_id INT NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    status ENUM('ACTIVE','CHECKED_OUT','ABANDONED') DEFAULT 'ACTIVE',
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
);
```

Cart_Items Table

```
CREATE TABLE Cart_Items (
    cart_item_id INT PRIMARY KEY AUTO_INCREMENT,
    cart_id INT NOT NULL,
    product_id INT NOT NULL,
    quantity INT NOT NULL CHECK (quantity > 0),
    price_at_time DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (cart_id) REFERENCES Cart(cart_id),
    FOREIGN KEY (product_id) REFERENCES Products(product_id)
);
```

3. Analysis Queries

Get all products in a category

```
SELECT p.product_id, p.product_name, p.price, c.category_name
FROM Products p
JOIN Categories c ON p.category_id = c.category_id
WHERE c.category_name = 'Smartphones';
```

Build category hierarchy (path view)

```
SELECT category_id, category_name, category_path
FROM Categories
ORDER BY category_path;
```

Detect outlier selections (surplus quantities)

```
SELECT
    ci.product_id,
    p.product_name,
    ci.quantity,
    (SELECT AVG(quantity) FROM Cart_Items WHERE product_id = ci.product_id) AS avg_qty,
    (SELECT STDDEV(quantity) FROM Cart_Items WHERE product_id = ci.product_id) AS stddev_qty
FROM Cart_Items ci
JOIN Products p ON ci.product_id = p.product_id
WHERE ci.quantity > (SELECT AVG(quantity) + 2 * STDDEV(quantity)
                     FROM Cart_Items WHERE product_id = ci.product_id);
```

4. Transaction Recovery and Reliability

All cart operations are enclosed within database transactions. In case of system failure, uncommitted transactions are rolled back. Use of **ACID** properties ensures data integrity (Atomicity, Consistency, Isolation, Durability). Backups and logging ensure recovery for e-commerce cart data.