Use Case-1: Building a Cart Analysis for Myph (Input & Output)

Input: Use Case Overview
Myph has launched a new phone range with an online cart system. Each product belongs to a category and customers can add products to their cart. We need to: Design an SQL data model to manage categories, products, and cart items. Implement queries for product selection, category-based browsing, and cart analytics. Detect outlier (surplus) selections in carts. Ensure transaction recovery in commerce operations.

Output: SQL Schema Implementation

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)
);
```

Example Input Data (INSERT Statements)

```
INSERT INTO Categories (category_name, category_path) VALUES
('Electronics', 'Electronics'),
('Phones', 'Electronics/Phones'),
('Smartphones', 'Electronics/Phones/Smartphones');

INSERT INTO Products (category_id, product_name, description, stock_quantity, price) VALUES
(3, 'Myph X1', 'Flagship smartphone', 100, 899.99),
(3, 'Myph X1 Pro', 'High-end smartphone', 50, 1099.99),
(3, 'Myph Lite', 'Budget smartphone', 200, 499.99);

INSERT INTO Customers (customer_name, email) VALUES
('Alice Johnson', 'alice@example.com'),
('Bob Smith', 'bob@example.com');

INSERT INTO Cart (customer_id, status) VALUES
(1, 'ACTIVE'),
(2, 'ACTIVE');

INSERT INTO Cart_Items (cart_id, product_id, quantity, price_at_time) VALUES
(1, 1, 2, 899.99),
(1, 2, 1, 1099.99),
(2, 3, 5, 499.99);
```

Example Output Queries and Results

Query 1: Products in 'Smartphones' 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';
```

product_id	product_name	price	category_name
1	Myph X1	899.99	Smartphones
2	Myph X1 Pro	1099.99	Smartphones
3	Myph Lite	499.99	Smartphones

Query 2: Detect Outlier (Surplus) Cart Selections

```
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);
```

product_id	product_name	quantity	avg_qty	stddev_qty
3	Myph Lite	5	2.6	1.2

Transaction Recovery and Commerce Reliability

All cart operations occur within ACID-compliant transactions. Uncommitted changes are rolled back if a failure occurs. Durability ensures completed transactions persist through logging and backups. Recovery tools (redo/undo logs) restore database to consistent state.