Retail Business Performance & Profitability Analysis

Objective: Analyze transactional retail data to uncover profit-draining categories, optimize inventory turnover, and identify seasonal product behavior. This deliverable includes: Tableau Dashboard, SQL queries, Python analysis scripts, and this PDF report with insights and strategic recommendations.

Data sources: transactions_clean (from raw_transactions), product_master, products

Key SQL Queries (excerpts)

```
-- retail_performance_queries.sql
-- 1) Import & cleaning (pseudo - adjust to your SQL dialect and table names)
/* Create cleaned transactions table from raw source */
CREATE TABLE transactions_clean AS
SELECT
  transaction_id,
  transaction_date::date AS transaction_date,
  customer_id,
  region,
  product_id,
  product_name,
  category,
  sub_category,
  quantity::integer AS quantity,
  unit_price::numeric AS unit_price,
  cost_price::numeric AS cost_price,
  COALESCE(discount, 0)::numeric AS discount,
  COALESCE(shipping_cost, 0)::numeric AS shipping_cost,
  COALESCE(inventory_days, NULL) AS inventory_days
FROM raw_transactions
WHERE transaction_date IS NOT NULL
  AND product_id IS NOT NULL
  AND unit_price IS NOT NULL
  AND cost_price IS NOT NULL;

-- 2) Basic profit calculations (row-level)
ALTER TABLE transactions_clean ADD COLUMN revenue numeric;
ALTER TABLE transactions_clean ADD COLUMN profit numeric;
UPDATE transactions_clean
SET revenue = (unit_price * quantity) - discount + shipping_cost,
    profit = (unit_price - cost_price) * quantity - discount - shipping_cost;

-- 3) Profit margin by category & sub_category
-- margin = total_profit / total_revenue
SELECT
  category,
  sub_category,
  SUM(profit) AS total_profit,
  SUM(revenue) AS total_revenue,
  CASE WHEN SUM(revenue) = 0 THEN 0 ELSE SUM(profit) / SUM(revenue) END AS profit_margin,
  SUM(quantity) AS units_sold
FROM transactions_clean
GROUP BY category, sub_category
ORDER BY profit_margin ASC;  -- lowest margins first (profit-draining)

-- 4) Inventory turnover / days vs profitability (aggregate at product level)
-- Assumes a product_master table with avg_inventory (units) or use inventory_days in transactions_clean
SELECT
  tc.product_id,
  tc.product_name,
  AVG(tc.inventory_days) AS avg_inventory_days,
  SUM(tc.profit) AS product_profit,
  SUM(tc.revenue) AS product_revenue,
  CASE WHEN SUM(tc.revenue)=0 THEN 0 ELSE SUM(tc.profit)/SUM(tc.revenue) END AS product_margin,
  SUM(tc.quantity) AS total_units_sold
FROM transactions_clean tc
GROUP BY tc.product_id, tc.product_name;

-- 5) Seasonal product behavior (month-level)
SELECT
```

Analysis Summary & Strategic Recommendations

1) Prioritize deep-dive into top 10 sub-categories with lowest profit margins. Consider price re-negotiation, bundle offers, or delisting low-margin SKUs.

2) For SKUs with avg_inventory_days > 60 and low sales: run promotions, consider markdowns, or transfer stock to higher-demand regions.

3) Use seasonal_trends to plan procurement: increase stock for high-season months and reduce purchase for off-season months.

4) Implement ABC classification by margin and turnover; focus working capital on A items.

5) Build Tableau dashboard with filters: region, product_type (category/sub_category), month/season, and inventory_days range.