SQL Data Analysis on Mobile Sales Data

I'll create a comprehensive SQL analysis of the mobile sales data provided. Here are the SQL queries I would run to extract meaningful insights from this dataset.

Basic Setup

First, I'll assume the data is loaded into a table called `mobile_sales` with the same structure as the CSV file.

```
-- 1. Basic Data Exploration
SELECT COUNT(*) AS total_transactions FROM mobile_sales;
SELECT DISTINCT Brand FROM mobile_sales;
SELECT MIN(Date) AS first_date, MAX(Date) AS last_date FROM mobile_sales;
```

Sales Analysis by Brand

```
-- 2. Total units sold by brand
SELECT
    Brand,
    SUM("Units Sold") AS total_units_sold,
    ROUND(SUM("Units Sold" * "Price Per Unit"), 2) AS total_revenue,
    ROUND(AVG("Price Per Unit"), 2) AS avg_price_per_unit
FROM mobile_sales
GROUP BY Brand
ORDER BY total_revenue DESC;

-- 3. Top 5 best-selling mobile models
SELECT
    "Mobile Model",
    Brand,
    SUM("Units Sold") AS total_units_sold,
    ROUND(SUM("Units Sold" * "Price Per Unit"), 2) AS total_revenue
FROM mobile_sales
GROUP BY "Mobile Model", Brand
ORDER BY total_units_sold DESC
LIMIT 5;
```

Time-Based Analysis

```
-- 4. Monthly sales trends
SELECT
    strftime('%Y-%m', Date) AS month,
    SUM("Units Sold") AS total_units,
    ROUND(SUM("Units Sold" * "Price Per Unit"), 2) AS monthly_revenue
FROM mobile_sales
GROUP BY strftime('%Y-%m', Date)
ORDER BY month;

-- 5. Daily sales performance
SELECT
    "Day Name",
    COUNT(*) AS transaction_count,
    SUM("Units Sold") AS total_units_sold,
    ROUND(SUM("Units Sold" * "Price Per Unit"), 2) AS daily_revenue
FROM mobile_sales
GROUP BY "Day Name"
ORDER BY daily_revenue DESC;
```

Customer Analysis

```
-- 6. Customer demographics by age group
SELECT
    CASE
        WHEN "Customer Age" < 20 THEN 'Under 20'
        WHEN "Customer Age" BETWEEN 20 AND 29 THEN '20-29'
        WHEN "Customer Age" BETWEEN 30 AND 39 THEN '30-39'
        WHEN "Customer Age" BETWEEN 40 AND 49 THEN '40-49'
        ELSE '50+'
    END AS age_group,
    COUNT(*) AS customer_count,
    ROUND(AVG("Customer Ratings"), 2) AS avg_rating
FROM mobile_sales
GROUP BY age_group
ORDER BY customer_count DESC;

-- 7. Top cities by sales
SELECT
    City,
    COUNT(*) AS transaction_count,
```

```
    ROUND(SUM("Units Sold" * "Price Per Unit"), 2) AS total_revenue
FROM mobile_sales
GROUP BY City
ORDER BY total_revenue DESC
LIMIT 10;
```

Payment Method Analysis

```
-- 8. Payment method preferences
SELECT
    "Payment Method",
    COUNT(*) AS transaction_count,
    ROUND(COUNT(*) * 100.0 / (SELECT COUNT(*) FROM mobile_sales), 2) AS
percentage,
    ROUND(AVG("Customer Ratings"), 2) AS avg_rating
FROM mobile_sales
GROUP BY "Payment Method"
ORDER BY transaction_count DESC;
```

Advanced Analysis with Subqueries and Joins

```
-- 9. Customers who made high-value purchases (top 10%)
SELECT
    "Customer Name",
    "Customer Age",
    City,
    "Mobile Model",
    "Units Sold",
    "Price Per Unit",
    ("Units Sold" * "Price Per Unit") AS transaction_value
FROM mobile_sales
WHERE ("Units Sold" * "Price Per Unit") > (
    SELECT PERCENTILE_CONT(0.9) WITHIN GROUP (ORDER BY ("Units Sold" * "Price Per
Unit"))
ORDER BY transaction_value DESC
LIMIT 20;

-- 10. Brand performance by city (using a view)
CREATE VIEW brand_city_performance AS
SELECT
    Brand,
```

```
    City,
    COUNT(*) AS transaction_count,
    SUM("Units Sold") AS total_units,
    ROUND(SUM("Units Sold" * "Price Per Unit"), 2) AS total_revenue
FROM mobile_sales
GROUP BY Brand, City;

-- Query the view for top performing brand-city combinations
SELECT * FROM brand_city_performance
ORDER BY total_revenue DESC
LIMIT 10;
```

Customer Satisfaction Analysis

```
-- 11. Customer ratings analysis by brand
SELECT
    Brand,
    COUNT(*) AS rating_count,
    ROUND(AVG("Customer Ratings"), 2) AS avg_rating,
    MIN("Customer Ratings") AS min_rating,
    MAX("Customer Ratings") AS max_rating
FROM mobile_sales
GROUP BY Brand
ORDER BY avg_rating DESC;

-- 12. Correlation between price and customer ratings
SELECT
    CASE
        WHEN "Price Per Unit" < 20000 THEN 'Budget (<20k)'
        WHEN "Price Per Unit" BETWEEN 20000 AND 50000 THEN 'Mid-range (20k-50k)'
        ELSE 'Premium (>50k)'
    END AS price_segment,
    ROUND(AVG("Customer Ratings"), 2) AS avg_rating,
    COUNT(*) AS transaction_count
FROM mobile_sales
GROUP BY price_segment
ORDER BY avg_rating DESC;
```

Optimization with Indexes

```
-- 13. Create indexes for performance optimization
CREATE INDEX idx_brand ON mobile_sales(Brand);
CREATE INDEX idx_date ON mobile_sales(Date);
CREATE INDEX idx_city ON mobile_sales(City);
CREATE INDEX idx_payment_method ON mobile_sales("Payment Method");
```

Seasonal Analysis

```
-- 14. Monthly sales by brand
SELECT
    Brand,
    strftime('%Y-%m', Date) AS month,
    SUM("Units Sold") AS monthly_units,
    ROUND(SUM("Units Sold" * "Price Per Unit"), 2) AS monthly_revenue
FROM mobile_sales
GROUP BY Brand, strftime('%Y-%m', Date)
ORDER BY Brand, month;

-- 15. Quarterly sales performance
SELECT
    Brand,
    CASE
        WHEN strftime('%m', Date) IN ('01','02','03') THEN 'Q1'
        WHEN strftime('%m', Date) IN ('04','05','06') THEN 'Q2'
        WHEN strftime('%m', Date) IN ('07','08','09') THEN 'Q3'
        ELSE 'Q4'
    END AS quarter,
    strftime('%Y', Date) AS year,
    SUM("Units Sold") AS quarterly_units,
    ROUND(SUM("Units Sold" * "Price Per Unit"), 2) AS quarterly_revenue
FROM mobile_sales
GROUP BY Brand, quarter, year
ORDER BY year, quarter, quarterly_revenue DESC;
```

These queries provide a comprehensive analysis of the mobile sales data, covering:

- Basic sales metrics by brand and model
- Time-based trends (daily, monthly, quarterly)
- Customer demographics and behavior

- Payment method preferences
- Customer satisfaction analysis
- High-value transaction identification
- Performance optimization with indexes

The views created can be used for further analysis without recomputing the aggregations each time.