BMW Sales SQL Project — Portfolio Report

Project Title: BMW Global Sales Analysis: Trends, Revenue Insights & Performance Dashboard

Purpose: Build a production-style SQL project that demonstrates data modeling, ETL-ready schema, advanced analytics (window functions, ranking, correlation), performance tuning, and a Power BI dashboard. This deliverable is portfolio-ready: include ER diagram, schema, sample queries (Basic → Advanced), insights, optimization notes, and a suggested dashboard layout.

---

1. Dataset Summary

Columns discovered in provided CSV: - `Model` (VARCHAR) - `Year` (INT) - `Region` (VARCHAR) - `Color` (VARCHAR) - `Fuel_Type` (VARCHAR) - `Transmission` (VARCHAR) - `Engine_Size_L` (FLOAT) - `Mileage_KM` (INT) - `Price_USD` (INT) - `Sales_Volume` (INT) - `Sales_Classification` (VARCHAR) — (High/Low/etc.)

Row count: ~50,000 (sample)

---

2. ER Diagram (text)

Note: For simplicity this project uses a star-ish normalized model split into `Models`, `Regions`, and fact table `Sales`.

---

3. Recommended Schema (DDL) — MySQL flavor

```sql
CREATE TABLE Models (
  model_id INT AUTO_INCREMENT PRIMARY KEY,
  model_name VARCHAR(100) UNIQUE NOT NULL,
  segment VARCHAR(50)
);

CREATE TABLE Regions (
  region_id INT AUTO_INCREMENT PRIMARY KEY,
  region_name VARCHAR(50) UNIQUE NOT NULL
);

CREATE TABLE Sales (
  sale_id INT AUTO_INCREMENT PRIMARY KEY,
  model_id INT NOT NULL,
  year INT NOT NULL,
  region_id INT NOT NULL,
  color VARCHAR(30),
  fuel_type VARCHAR(30),
  transmission VARCHAR(30),
  engine_size_l DECIMAL(3,1),
  mileage_km INT,
  price_usd INT,
  sales_volume INT,
  sales_classification VARCHAR(20),
  FOREIGN KEY (model_id) REFERENCES Models(model_id),
  FOREIGN KEY (region_id) REFERENCES Regions(region_id)
);

-- Indexes for performance
CREATE INDEX idx_sales_year ON Sales(year);
CREATE INDEX idx_sales_model ON Sales(model_id);
CREATE INDEX idx_sales_region ON Sales(region_id);
```

ETL tips: Load distinct `Model` and `Region` first, then map and insert to `Sales` with FK ids. Clean `Price_USD` and `Mileage_KM` outliers and ensure `Year` types.

---

4. Queries — Basic → Intermediate → Advanced

(Full list included; highlight: advanced queries use window functions, correlation, ranking, and views.)

Basic (examples)

```sql
-- 1. Distinct models
SELECT DISTINCT model_name FROM Models;

-- 2. Total cars sold in 2024
SELECT SUM(sales_volume) FROM Sales WHERE year = 2024;

-- 3. Top 5 most expensive cars
SELECT m.model_name, s.price_usd
FROM Sales s JOIN Models m ON s.model_id = m.model_id
ORDER BY s.price_usd DESC LIMIT 5;
```

Intermediate (examples)

```sql
-- Average price by model
SELECT m.model_name, ROUND(AVG(s.price_usd),2) AS avg_price
FROM Sales s JOIN Models m USING(model_id)
GROUP BY m.model_name;

-- Total sales by year
SELECT year, SUM(sales_volume) AS total_sales
FROM Sales GROUP BY year ORDER BY total_sales DESC;
```

Advanced (selected, production-ready queries)

```sql
-- Cumulative Sales by Year (running total)
SELECT year,
       SUM(sales_volume) AS total_sales,
       SUM(SUM(sales_volume)) OVER (ORDER BY year) AS cumulative_sales
FROM Sales
GROUP BY year
ORDER BY year;

-- Rank models by total sales
SELECT model_name, total_sales,
       RANK() OVER (ORDER BY total_sales DESC) AS sales_rank
FROM (
  SELECT m.model_name, SUM(s.sales_volume) AS total_sales
  FROM Sales s JOIN Models m USING(model_id)
  GROUP BY m.model_name
) x;

-- Percentage contribution by region
```

```sql
SELECT r.region_name,
       SUM(s.sales_volume) AS region_sales,
       ROUND( SUM(s.sales_volume) * 100.0 / SUM(SUM(s.sales_volume)) OVER (),
2) AS pct_contribution
FROM Sales s JOIN Regions r USING(region_id)
GROUP BY r.region_name
ORDER BY pct_contribution DESC;

-- Top-selling model per region
SELECT region_name, model_name, total_sales FROM (
  SELECT r.region_name, m.model_name, SUM(s.sales_volume) AS total_sales,
         RANK() OVER (PARTITION BY r.region_id ORDER BY SUM(s.sales_volume)
DESC) AS rk
  FROM Sales s JOIN Regions r USING(region_id)
               JOIN Models m USING(model_id)
  GROUP BY r.region_name, m.model_name, r.region_id
) t WHERE rk = 1;

-- YoY growth %
SELECT year, total_sales,
       LAG(total_sales) OVER (ORDER BY year) AS prev_sales,
       ROUND((total_sales - LAG(total_sales) OVER (ORDER BY year)) *100.0 /
LAG(total_sales) OVER (ORDER BY year),2) AS yoy_pct
FROM (
  SELECT year, SUM(sales_volume) AS total_sales FROM Sales GROUP BY year
) y ORDER BY year;

-- Correlation Engine_Size_L vs Price_USD
SELECT (COUNT(*) * SUM(engine_size_l * price_usd) -
SUM(engine_size_l)*SUM(price_usd)) /
       SQRT((COUNT(*)*SUM(POWER(engine_size_l,2)) - POWER(SUM(engine_size_l),
2)) *
            (COUNT(*)*SUM(POWER(price_usd,2)) - POWER(SUM(price_usd),2))) AS
correlation
FROM Sales;

-- View: price vs model average
CREATE VIEW ModelPriceComparison AS
SELECT s.*, m.model_name,
       CASE WHEN s.price_usd > AVG(s.price_usd) OVER (PARTITION BY s.model_id)
THEN 'Above Average' ELSE 'Below Average' END AS price_status
FROM Sales s JOIN Models m USING(model_id);
```

---

5. Insights & Example Findings (what to look for)

- Top models contributing to >50% of global sales
- Regions with fastest YoY growth (and possible market expansion opportunities)
- Price elasticity: price vs sales_volume by model/region
- Correlation sign between engine size and price — useful to justify pricing tiers
- Sales classification distribution (`High` vs `Low`) and drivers (model, region, fuel)

---

6. Performance & Production Notes

- Add indexing on `year`, `model_id`, `region_id`, and `price_usd` for analytic queries.
- Use materialized summary tables for heavy dashboards (e.g., `monthly_sales_summary`) refreshed nightly.
- Use `EXPLAIN` to validate query plans and add composite indexes where necessary.

---

7. Power BI Dashboard Structure (recommended visuals)

1. **KPI cards:** Total Sales, YoY Growth %, Avg Price, Top Region
2. **Time series:** Total Sales by Year (with running total overlay)
3. **Bar chart:** Sales by Model (top 10) with drill-through
4. **Map / Filled map:** Sales by Region (percentage contribution)
5. **Scatter plot:** Engine_Size_L vs Price_USD (size by Sales_Volume)
6. **Table:** Top-selling model per region with conditional formatting
7. **Filters / Slicers:** Year, Region, Fuel_Type, Transmission, Model

---

8. Project Deliverables (what to include in portfolio)

- `README.md` (project overview, dataset source, steps to run)
- DDL scripts (create tables, indexes)
- ETL notebook (Python / SQL) showing data cleaning and loading
- `queries.sql` (organized: basic → intermediate → advanced)
- Power BI file (`.pbix`) or screenshots
- Short write-up with 5 key business insights and recommendations

---

9. Next Steps I can do for you

- Generate full `queries.sql` file with all 20 queries.
- Create the ETL Python notebook to load CSV into the schema and handle FK mapping.
- Build the Power BI layout and export visuals or publish screenshots.

Tell me which of the next steps you want me to build first.