Business Overview

- **Name:** Volta Motors
- **What it does:** Designs, manufactures, and sells premium electric vehicles (EVs) directly to consumers globally. They focus on innovation, performance, and sustainable manufacturing.
- **Target Customers:** Mid-to-high-end consumers who value technology, performance, and environmental sustainability. Customers are typically tech-savvy individuals or families with disposable income.
- **Revenue Streams:**
    1. **Primary: EV Sales** (Direct-to-Consumer model).
    2. **Secondary: Maintenance and Service** (post-sale servicing, part replacements).
    3. **Others: Charging Station Network** fees and subscriptions.
- **Global Operations:** Manufacturing sites in North America, South America, Europe, Asia (excl. China), and China.

Data Needs and Decision-Making

The operations of Volta Motors can be significantly improved by data. Below is a breakdown of the key data that would be important to store and use:

I. Internal Data for Decision Making

- **Manufacturing & Supply Chain Data:**
    - **Decision:** How many parts (e.g. battery cells, microchips) should be ordered from which supplier for the next quarter?
    - **Required Data:**
        - **Supplier Performance:** Supplier IDs, part costs, lead times, quality ratings, and historical on-time delivery percentages.
        - **Inventory Levels:** Current stock of raw materials and finished components at each manufacturing plant.
        - **Production Metrics: Yield rate** (the percentage of successful products vs. total started) by plant and by production line, defect codes, and time-to-manufacture per vehicle model.

- **Sales & Demand Forecasting Data:**

    - **Decision:** What level of finished-goods inventory needs to be kept to meet demand in the peak season?
    - **Required Data:**

        - **Order Backlog:** Detailed data on pre-orders, configuration choices (color, features, battery pack), and estimated delivery dates.
        - **Historical Sales:** Sales volume by model, region, month, and price point.
        - **Website/App Traffic:** Customer interest data (e.g. number of car configurations started but not completed).

II. External/User-Facing Data

This data is either presented to the customer or relates directly to their transactions:

- **Customer & Sales Data:**
    - **Required Data:** Customer profiles (name, address, contact info), vehicle VINs, purchase dates, warranty start/end dates, and service history.
    - **Use Case:** Customers need to see their vehicle's service schedule, and sales teams need to track lead status and commission.
- **Charging Network Data:**
    - **Required Data:** Location, type, and usage data for Volta Motors charging stations. Customer charging session details (start/end time, kWh delivered, cost).
    - **Use Case:** Reporting energy consumption to regulatory bodies and billing the customer.

III. Record-Keeping & Compliance Data

This data is essential for financial, legal, and regulatory obligations:

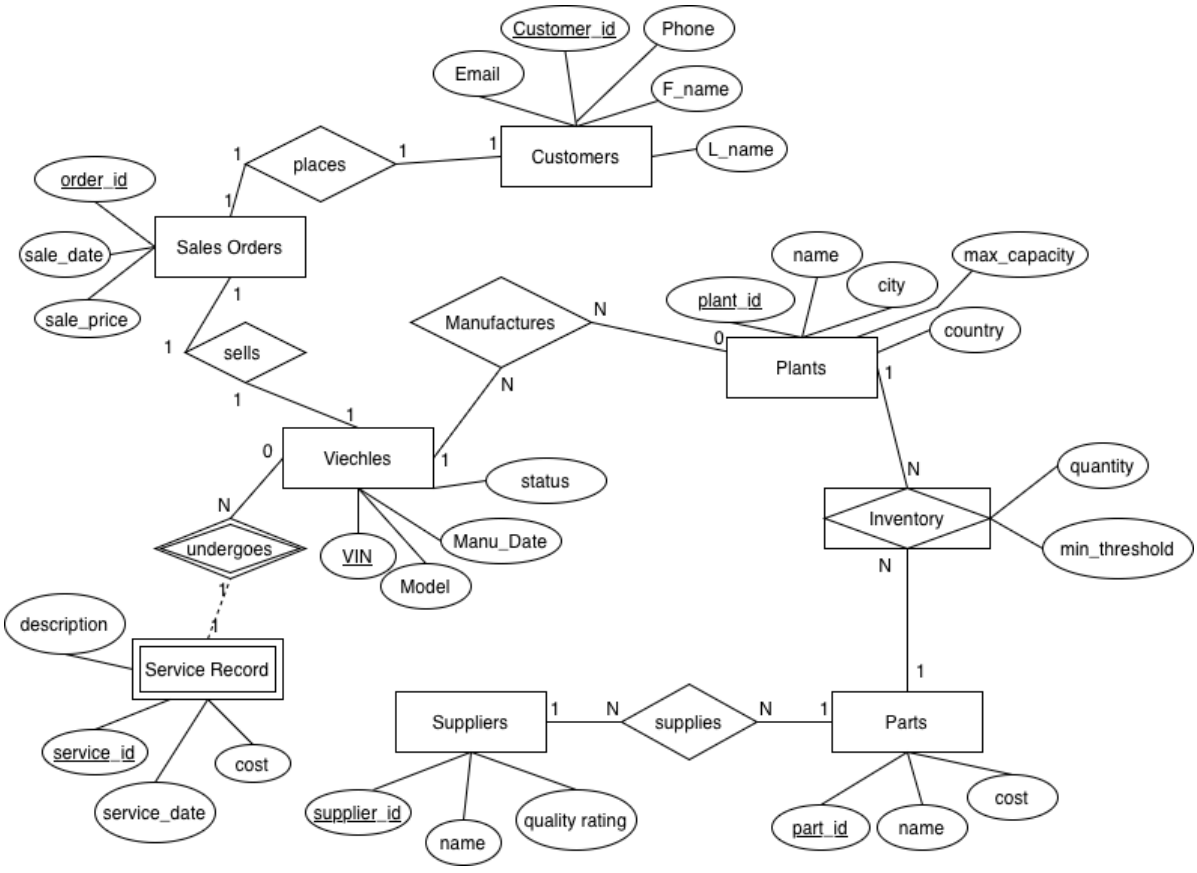
- **Financial & Accounting Data:** Revenue recognition from sales, cost of goods sold (COGS) by plant, and payroll data.
    - **Use Case:** Quarterly and annual financial reporting (reporting) and tax compliance.
- **Employee Performance Data:** Sales representative performance (quota attainment, customer satisfaction scores), technician efficiency, and manufacturing team metrics.
    - **Use Case:** Compensation calculations (record-keeping) and identifying training needs (reporting).
- **Regulatory & Quality Data:** Detailed logs of manufacturing quality data (e.g. safety checks, component test results) tied to specific VINs.
    - **Use Case: Compliance** with local safety standards (e.g., NHTSA in the US, UNECE in Europe) and managing product recalls if defects are found.

Data Description

To support the operations of Volta Motors, the database track the following core entities:

- **Manufacturing & Supply:**
  - **Plants:** The physical locations where cars are built.
  - **Suppliers:** External partners providing components.
  - **Parts:** The catalog of components (batteries, chips, tires).
  - **Inventory:** The association between a Plant and Parts (how many of Part X are at Plant Y).
- **Core Product:**
  - **Vehicles:** Identify each unit by a unique VIN, model, manufacturing date, and current status (e.g. 'In Production', 'Ready', 'Sold').
- **Sales & Service:**
  - **Customers:** Personal and contact info of buyers.
  - **Sales Orders:** The transaction record linking a Customer to a specific Vehicle.
  - **Service Records:** Logs of maintenance performed on vehicles post-sale.

Entity-Relationship (ER) Diagram



Relational Schema

Here is the schema converted from the diagram. I have indicated Primary Keys (PK) and Foreign Keys (FK).

1. Plants (plant_id [PK], name, city, country, max_capacity)
2. Suppliers (supplier_id [PK], name, quality_rating)
3. Parts (part_id [PK], name, cost, supplier_id [FK])
4. Inventory (plant_id [FK], part_id [FK], quantity, min_threshold)
5. Vehicles (vin [PK], model_name, manufacture_date, status, plant_id [FK])
6. Customers (customer_id [PK], first_name, last_name, email, phone)
7. SalesOrders (order_id [PK], customer_id [FK], vin [FK], sale_date, sale_price)
8. ServiceRecords (service_id [PK], vin [FK], service_date, description, cost)

Normalization Analysis

The schema designed above is largely in 3NF. But there is one specific area where I choose to denormalize: SalesOrders. In the SalesOrders table, I included sale_price because prices change over time. If I only linked to a central "Model Price," and that price increased in 2026, all historical records from 2025 would appear to have sold for the 2026 price. By storing the sale_price directly in the Order table, I preserve the historical accuracy of the transaction.

Database Analysis Questions

Which relation would likely see the most UPDATE operations? Inventory. Every time a vehicle moves along the assembly line, parts are consumed. If Volta Motors produces thousands of cars a day, the inventory counts for screws, battery cells, and chips must be decremented constantly. Additionally, as new shipments arrive from suppliers, these numbers must be incremented.

Which table would likely see the fewest UPDATE operations? SalesOrders. Once a car is sold, the date, price, and customer associated with that Order ID are historical facts that do not change.

In which table will adherence to ACID properties be the most important? Inventory. If two manufacturing lines in the same plant are trying to grab the last batch of microchips at the same time, without ACID properties both lines might read that chips are available. The database would record a negative inventory or a conflict. ACID ensures that if Line A takes the chips, the transaction is processed fully before Line B can read the quantity.

Two Integrity Constraints:

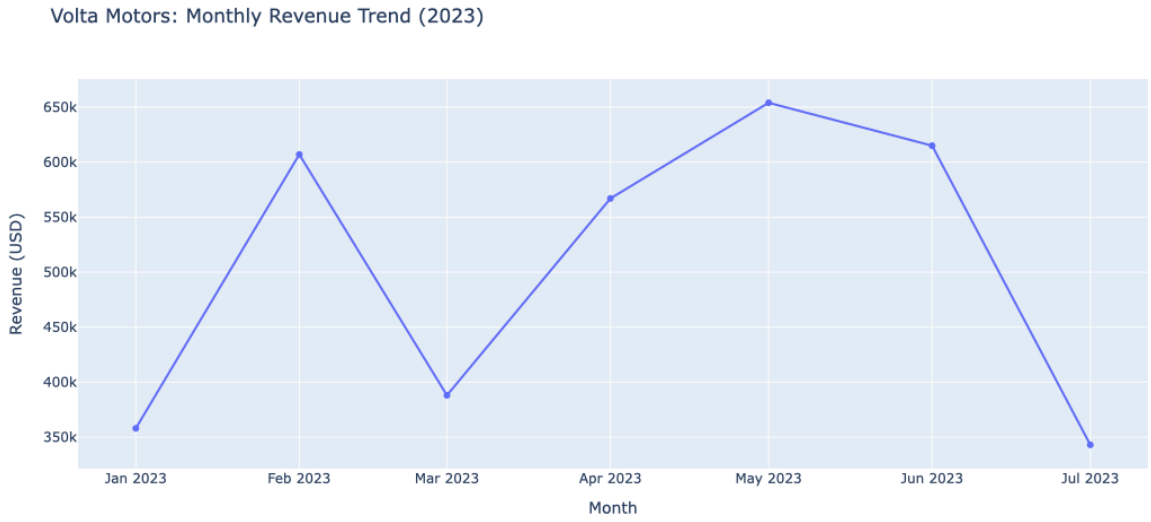
1. **Inventory Non-Negativity :**
    - Logic: You cannot have fewer than zero parts.
    - Importance: Prevents logic errors in the manufacturing planning algorithms.
2. **Vehicle Status Validity :**
    - Logic: A car cannot be in a state that doesn't exist.
    - Importance: Ensures that reporting dashboards don't break due to typos.

Queries and Plot

1. Monthly Revenue Trend

- **SQL Query: Extracts the sale date converted into a "Year-Month" string format and sums the `sale_price` for all orders within that month.**

  SELECT TO_CHAR(sale_date, 'YYYY-MM') AS month_year, SUM(sale_price) AS total_revenue
  FROM SalesOrdersGROUP BY TO_CHAR(sale_date, 'YYYY-MM') ORDER BY month_year;

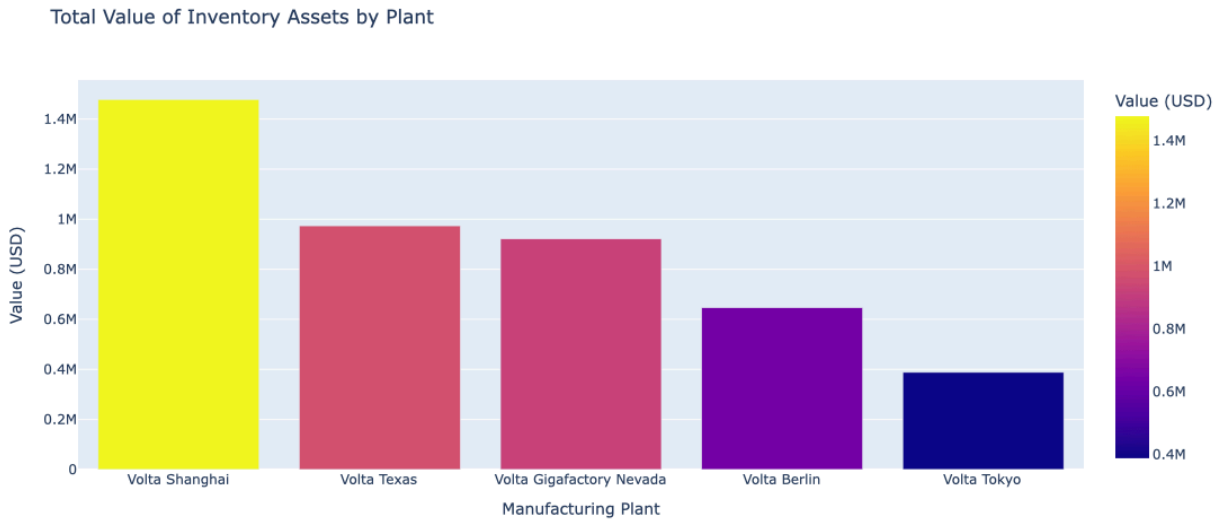


2. Inventory Asset Value by Plant

- **SQL Query: Joins the *Inventory, Parts, and Plants* tables. It calculates the dollar value of stock for every single part (`quantity` *`cost`)**

  SELECT pl.name AS plant_name, SUM(i.quantity * p.cost) AS total_inventory_value
  FROM Inventory i
  JOIN Parts p ON i.part_id = p.part_id
  JOIN Plants pl ON i.plant_id = pl.plant_id

GROUP BY pl.name
ORDER BY total_inventory_value DESC;



3. Revenue Share by Vehicle Model

- **SQL Query: Links *SalesOrders* to *Vehicles* to find out which car model was actually sold. It groups the results by `model_name` (Model S, Model 3, etc.)**

  SELECT v.model_name, COUNT(s.order_id) AS units_sold, SUM(s.sale_price) AS total_revenue
  FROM SalesOrders s
  JOIN Vehicles v ON s.vin = v.vin
  GROUP BY v.model_name;