# **📘 AutoPartsPro Schema Rationale and ER Diagram (Updated)**

## **🎯 Purpose**

This document explains the purpose of each table in the AutoPartsPro schema, the design rationale behind it, and how it relates to other tables. It also includes a complete dbdiagram.io-compatible schema.

## **📦 Tables and Rationale**

### **customers**

Stores basic customer data.

* Relations: vehicles.customer\_id → customers.id, customer\_orders.customer\_id → customers.id

### **vehicle\_types**

Reference table for reusable vehicle configurations (make, model, year, version).

* Relations: vehicles.vehicle\_type\_id → vehicle\_types.id

### **vehicles**

Represents a physical vehicle owned by a customer.

* Includes license\_plate and optional vin.
* Relations: customer\_id, vehicle\_type\_id, customer\_orders.vehicle\_id

### **skus**

Standard part identifiers that group interchangeable part instances.

* Includes size and description at SKU level.
* Relations: used by parts, job\_parts, inventory

### **parts**

Individual part instances with creation date and inventory status.

* Now simplified to reflect instantiation only.
* Relations: sku\_id, used by part\_market\_data, vehicle\_parts\_catalog

### **part\_market\_data**

Tracks economic data for each specific part.

* Relations: part\_id

### **misc\_settings**

System-wide constants for cost calculation and warehouse assumptions.

### **inventory\_constraints**

Represents reusable constraints for storage capacity and cost.

### **inventory**

Tracks quantity and date of entry for each SKU in stock.

* Relations: sku\_id, inventory\_constraints\_id

### **vehicle\_parts\_catalog**

Maps compatible parts to vehicle types.

* Composite PK: (vehicle\_type\_id, part\_id)

### **customer\_orders**

Captures the customer’s description of what needs service.

* Relations: customer\_id, vehicle\_id

### **workshop\_orders**

Represents the workshop’s formal repair/service response.

* Relations: customer\_order\_id, order\_jobs.order\_id

### **jobs**

Represents a unit of work to be performed.

* Reverted to use inline job\_type enum instead of FK
* Relations: used in order\_jobs, job\_parts

### **order\_jobs**

Join table between workshop\_orders and jobs

* Useful for many-to-many support

### **job\_parts**

Links jobs to the SKUs they consume.

* Relations: job\_id, sku\_id

### **profit\_loss**

Tracks financial performance per workshop order.

* Relations: order\_id → workshop\_orders.id