**To implement the Garage Inventory and Order Management System, we can design the database schema with proper relational mappings. Below is the database table structure, including relationships between tables for the Inventory, Order, and Supplier services.**

1. **Inventory Table**

This table stores the details of the parts in the garage's inventory.

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| inventory\_id | BIGINT | PRIMARY KEY | Unique ID for each part. |
| name | VARCHAR(255) | NOT NULL | Name of the part. |
| available\_qty | INT | NOT NULL | Current quantity available in stock. |
| threshold\_limit | INT | NOT NULL | Minimum stock level before triggering re-order. |
| min\_order\_qty | INT | NOT NULL | Minimum order quantity to restock. |
| supplier\_id | BIGINT | FOREIGN KEY | References the supplier\_id from supplier table. |

**Table Definition**

CREATE TABLE inventory (

inventory\_id BIGINT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

available\_qty INT NOT NULL,

threshold\_limit INT NOT NULL,

min\_order\_qty INT NOT NULL,

supplier\_id BIGINT,

FOREIGN KEY (supplier\_id) REFERENCES supplier(supplier\_id)

);

1. **Supplier Table**

This table stores information about the suppliers (Supplier-A and Supplier-B).

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| supplier\_id | BIGINT | PRIMARY KEY | Unique ID for the supplier. |
| name | VARCHAR(255) | NOT NULL | Name of the supplier (Supplier-A/B). |
| type | VARCHAR(50) | NOT NULL | Supplier type (Local/International). |

**Table Definition**

CREATE TABLE supplier (

supplier\_id BIGINT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

type VARCHAR(50) NOT NULL

);

1. **Order Table**

This table stores information about each order placed to suppliers.

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| order\_id | BIGINT | PRIMARY KEY | Unique ID for each order. |
| inventory\_id | BIGINT | FOREIGN KEY | References the part\_id from inventory. |
| quantity | INT | NOT NULL | Quantity of the parts ordered. |
| order\_time | TIMESTAMP | NOT NULL | Time when the order was placed. |
| supplier\_id | BIGINT | FOREIGN KEY | References the supplier\_id from supplier. |
| status | VARCHAR(50) | NOT NULL | Status of the order (PENDING, PLACED, etc.). |

**Table Definition**

CREATE TABLE orders (

order\_id BIGINT PRIMARY KEY,

part\_id BIGINT,

quantity INT NOT NULL,

order\_time TIMESTAMP NOT NULL,

supplier\_id BIGINT,

status VARCHAR(50) NOT NULL,

FOREIGN KEY (inventory\_id) REFERENCES inventory(inventory\_id),

FOREIGN KEY (supplier\_id) REFERENCES supplier(supplier\_id)

);

1. **Table Relations**
2. **One-to-Many (Supplier to Inventory Parts)**:

* One supplier can provide many inventory parts, but each part is provided by a single supplier.
* The supplier\_id column in the inventory table references the supplier\_id in the supplier table.

1. **One-to-Many (Inventory Part to Orders)**:

* One Inventory part can have multiple orders, but each order is placed for a specific part.
* The inventory\_id column in the orders table references the inventory\_id in the inventory table.

1. **One-to-Many (Supplier to Orders)**:

* One supplier can receive multiple orders, but each order is placed with a single supplier.
* The supplier\_id column in the orders table references the supplier\_id in the supplier table.

**ER Diagram**

Here’s a summary of how the tables are related:

supplier (1) --------- (N) inventory

\ |

\ |

\ |

\ |

\-----------(N) orders

**supplier to inventory**: One supplier provides multiple parts.

**inventory to orders**: One part can be ordered multiple times.

**supplier to orders**: One supplier receives multiple orders.

**Sample Data**

1. **Supplier Table** (supplier)

| **supplier\_id** | **name** | **type** |
| --- | --- | --- |
| 1 | Supplier-A | Local |
| 2 | Supplier-B | International |

1. **Inventory Table (inventory)**

| **part\_id** | **name** | **available\_qty** | **threshold\_limit** | **min\_order\_qty** | **supplier\_id** |
| --- | --- | --- | --- | --- | --- |
| **101** | **Brake Pad** | **10** | **5** | **20** | **1** |
| **102** | **Clutch Plate** | **5** | **5** | **15** | **2** |

1. **Order Table** (orders)

| **order\_id** | **part\_id** | **quantity** | **order\_time** | **supplier\_id** | **status** |
| --- | --- | --- | --- | --- | --- |
| 10001 | 101 | 20 | 2024-11-02 00:15:00 | 1 | PLACED |
| 10002 | 102 | 15 | 2024-11-02 00:30:00 | 2 | PENDING |

**Indexes**

To optimize the performance of queries:

1. **Index on available\_qty in inventory**: For efficient lookups when checking low stock levels

**CREATE INDEX idx\_available\_qty ON inventory(available\_qty);**

1. **Index on order\_time in orders:** For efficient order scheduling and auditing

**CREATE INDEX idx\_order\_time ON orders(order\_time);**

**This database structure, following the one-to-many relationships and indexing, ensures smooth functioning of the automated inventory and order management system. The structure supports the core requirements such as managing parts, suppliers, and placing automated orders efficiently.**