

Designing Data Model for Speedy Gro

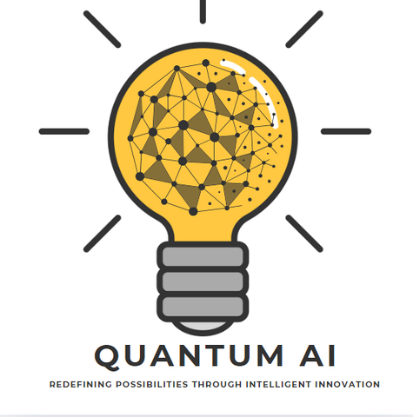
TEAM MEMBERS

AKASH C0910253

JAGADEESH C0912884

SAKSHI C0908000

PRANIT C0913039



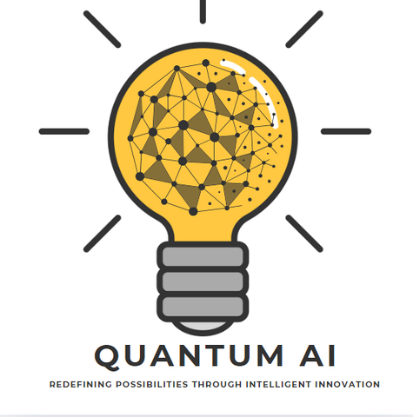
### **Report: Conceptual and Logical Data Model for Speedy Gro’s Operational Needs**

#### **1. Introduction**

**Purpose of the Report**

This report aims to present and recommend a conceptual data model and logical data model designed to meet the operational needs of Speedy Gro, a delivery service provider. Speedy Gro specializes in delivering a variety of products efficiently and reliably. To achieve its operational goals, the company requires a structured, scalable, and comprehensive data model that connects the various elements of its business operations.

The report will define the essential data entities, their relationships, and the ways in which they interact to manage critical functions within the company. The proposed data model will play a key role in tracking and optimizing processes related to customers, orders, warehouses, and delivery drivers. By establishing clear connections between these components, Speedy Gro will be able to integrate its operations smoothly and enhance overall process efficiency.



**Overview of Speedy Gro**

Speedy Gro is a company focused on providing fast and reliable delivery services. Their operations depend heavily on maintaining accurate data regarding customer orders, the availability of delivery drivers, and warehouse inventory levels. This data model will help streamline these operations. Speedy Gro is a fast-growing grocery delivery service with the ambition of providing an efficient shopping experience for its customers. Along with rapid growth, some of the issues it encounters are customer order management, driver tracking, and keeping track of inventory across different warehouses. In the following section, we will design a data model to organize and link major business components that help us solve these problems. The model ensures that each part of the system functions properly to allow effective operations management at Speedy Gro. The organization will then have a planned direction towards a heightened level of customer satisfaction and make wiser business decisions with real-time information.

#### **2. Understanding the Business Process**

**Operational Requirements**

The overall operational processes for Speedy Gro include:  
Customer Orders: Customers place orders for grocery items over an o

A light bulb with lines and dots

AI-generated content may be incorrect.

online platform, and such orders are delivered at addresses scheduled in advance.  
Warehouses: Products are warehoused in a variety of warehouse locations, and for each such location, proper tracking of inventories and stocks is a necessity.

**Delivery Drivers:**

There are delivery drivers for single orders, and for each such delivery, routes, and delivery times must be optimized for both cost and time.  
Such operations depend upon proper and timely information. Information models must allow such requirements through proper management of orders for customers, tracking delivery. The company handles transportation management, monitors storage spaces, tracks performance figures and maintains driver data.

**Data Requirements and Challenges in Operations**

1. Our different warehouse points depend on us checking product quantity levels and moving stock between sites.

2. The delivery drivers follow special assignments and move orders based on optimal route and scheduling plans. Basic physical operations depend on getting correct information immediately.

The data model should create systems that help run customer orders smoothly alongside tracking delivery schedules and managing warehouse supplies while evaluating driver results.

**Data Needs and Operational Challenges**

To attain operational efficiency, real-time information in three areas will be necessitated for Speedy Gro:

A light bulb with lines and dots

AI-generated content may be incorrect.

**Customer orders and tracking:** Having updated information about orders' state (e.g., pending, packed, shipped, delivered) will matter. The system will have to maintain updated information about orders, including delivery state and received feedback.

**Driver routing and assignment:** Information about availability of drivers, dispatched orders, and most efficient routes for delivery will have to be kept in view in order to maintain efficiency and maximize productivity.

**Warehouse inventory management**: Real-time information about current inventories and availability in warehouses will have to be kept in track in order to deliver orders in a timely manner.

The system will have to scale with increased volumes of orders, new drivers, and new locations for warehouses. It will have to be flexible enough to adapt with increased requirements for Speedy Gro over a period of time.

#### **3. Conceptual Data Model Overview**

**Definition of a Conceptual Data Model**

A light bulb with lines and dots

AI-generated content may be incorrect.

A conceptual data model provides an abstract representation of the data entities and their relationships within a system. It helps to visualize how data flows across different components of the system without focusing on technical details.

**High-Level Conceptual Model**

The conceptual data model will include the following entities:

* **Customer**: Represents an individual or business placing orders.
* **Delivery Driver**: Represents the employees who deliver the products.
* **Warehouse**: Represents storage locations for inventory.
* **Order**: Represents a customer’s purchase and associated products.

The relationships between these entities are as follows:

* A **customer** places one or more **orders**.
* An **order** is processed by one or more **warehouses**.
* An **order** is assigned to a **delivery driver** for delivery.
* A **warehouse** stores multiple **products**, which are linked to **orders**.

A light bulb with lines and dots

AI-generated content may be incorrect.

#### **4. Entities in the Data Model**

**Customers**

Attributes:

* Customer\_ID (Primary Key)
* Name
* Address
* Phone Number
* Email
* Payment Method

**Delivery Drivers Attributes:**

* Driver\_ID (Primary Key)
* Name
* Vehicle\_ID
* Phone Number
* Shift (Schedule)
* Assigned\_Orders (Foreign Key to Orders)

**Warehouses**

A light bulb with lines and dots

AI-generated content may be incorrect.

Attributes:

* Warehouse\_ID (Primary Key)
* Location
* Capacity
* Products\_in\_stock (Foreign Key to Products)

**Orders**

Attributes:

* Order\_ID (Primary Key)
* Customer\_ID (Foreign Key to Customers)
* Warehouse\_ID (Foreign Key to Warehouses)
* Order\_Date
* Delivery\_Date

Delivery\_Status

* Total\_Amount

**Products**

A light bulb with lines and dots

AI-generated content may be incorrect.

Attributes:

* Product\_ID (Primary Key)
* Product\_Name
* Quantity\_Available
* Price
* Warehouse\_ID (Foreign Key to Warehouses)

### **5. Entity Relationships and Data Flow**

#### **1. Customer-Order Relationship (One-to-Many)**

* **Overview**:
* The **Customer-Order** relationship is classified as **one-to-many (1: N)**. This means that a **single Customer** can place **multiple Orders**, but each **Order** is tied to only one **Customer**.
* **Details**:

A light bulb with lines and dots

AI-generated content may be incorrect.

* + A **Customer** may place numerous orders over time. For instance, a customer could make different purchases on separate occasions.
  + Every **Order** is linked to a specific **Customer**. The **Customer\_ID** within the **Order** table connects each order to its respective customer.
* **Relationship Type**:
  + **One-to-Many (1:N)**: A **Customer** can place multiple **Orders**, but every **Order** is linked to a single **Customer**.
* **Illustrative Example**:
  + **Customer 101** places **Order 001**, **Order 002**, and **Order 003**.
  + These orders are all associated with **Customer 101**, and no other customer can share these specific orders.

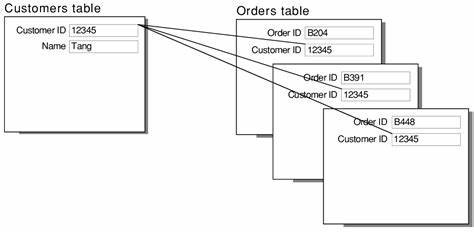


Figure 1 (One to many relation)

#### A light bulb with lines and dots AI-generated content may be incorrect.

#### **2. Order-Warehouse Relationship (Many-to-One)**

* **Overview**:

The **Order-Warehouse** relationship is defined as **many-to-one (N:1)**. This implies that each **Order** is fulfilled by one **Warehouse**, but a **Warehouse** can fulfill multiple **Orders**.

* **Details**:
  + Several **Orders** might be processed by the same **Warehouse**, meaning different customers could place orders that are all handled by the same warehouse.
  + However, each **Order** can only be processed by one warehouse since order fulfillment depends on stock available at a particular location.
* **Relationship Type**:
  + **Many-to-One (N:1)**: A **Warehouse** may handle multiple **Orders**, but each **Order** is associated with one **Warehouse**.
* **Illustrative Example**:
  + **Order 001**, **Order 002**, and **Order 003** could all be processed by **Warehouse 1**.
  + **Order 004** might be fulfilled by **Warehouse 2**.
  + This shows that while a **Warehouse** can handle multiple orders, each **Order** is assigned to only one warehouse.

A light bulb with lines and dots

AI-generated content may be incorrect.

#### **3. Order-Delivery Driver Relationship (One-to-One or One-to-Many)**

* **Overview**:

The relationship between **Orders** and **Delivery Drivers** could either be **one-to-one (1:1)** or **one-to-many (1: N)**, depending on the operational structure and delivery requirements.

* **Details**:
  + **One-to-One (1:1)**: In a few cases, one **Delivery Driver** is dispatched for one order, and each driver is responsible for a particular delivery.
  + **One-to-Many (1: N)**: In some scenarios, multiple **Orders** are handled by single **Delivery Driver** at once, such as when delivering multiple packages on the same route.
* **Relationship Type**:
  + **One-to-One (1:1)** or **One-to-Many (1: N)**: It is evident that one **Order** can be linked to a particular **Delivery Driver**, but **Delivery Driver is able to** manage numerous orders.
* **Illustrative Example**:
  + **Order 001** is assigned to **Driver A**, while **Order 002** is assigned to **Driver B** (One-to-One).

A light bulb with lines and dots

AI-generated content may be incorrect.

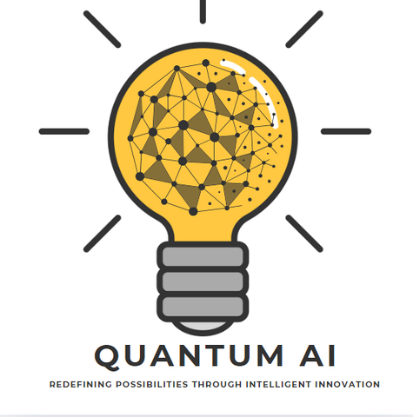
* + Alternatively, **Driver A** might be assigned **Order 001**, **Order 002**, and **Order 003** (One-to-Many).

#### **4. Warehouse-Product Relationship (One-to-Many)**

* **Overview**:

The **Warehouse-Product** relationship is **one-to-many (1: N)**, meaning that a **Warehouse** can store many **Products**, but each **Product** is linked to only one **Warehouse** at any given time.

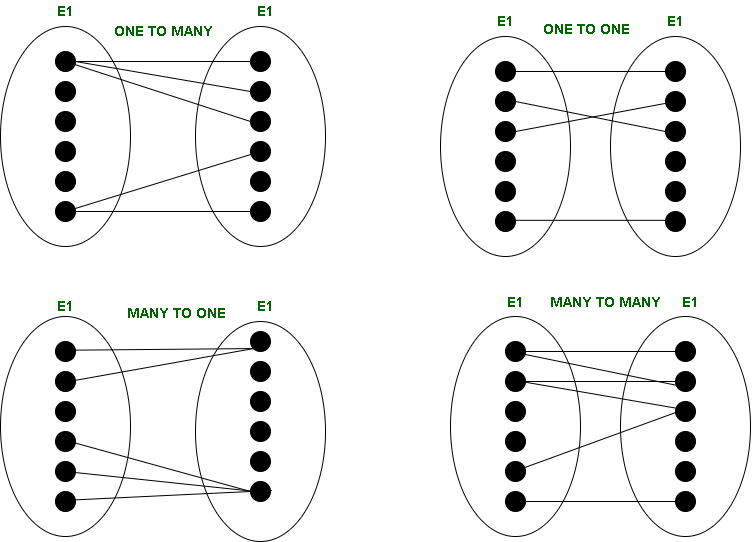
* **Details**:
  + A **Warehouse** can house multiple different types of **Products**.
  + Each **Product** is stored in only one **Warehouse**, though that warehouse may hold many different items.
* **Relationship Type**:
  + **One-to-Many (1: N)**: A **Warehouse** can stock multiple **Products**, but each **Product** is affiliated with one **Warehouse**.
* **Illustrative Example**:
  + **Warehouse 1** might store **Product A**, **Product B**, and **Product C**.
  + **Warehouse 2** might contain **Product D** and **Product E**.
  + **Product A** is specifically stored in **Warehouse 1** and cannot be found in any other warehouse.

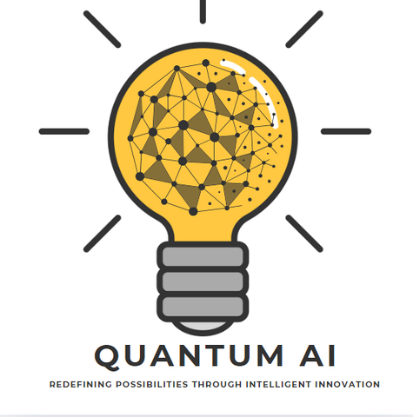


### **Summary of Relationships and Cardinality**

|  |  |  |  |
| --- | --- | --- | --- |
| **Relationship** | **Entities** | **Cardinality** | **Description** |
| **Customer-Order** | Customer to Order | One-to-Many | A **Customer** can place many **Orders**. |
| **Order-Warehouse** | Order to Warehouse | Many-to-One | An **Order** is fulfilled by one **Warehouse**, but a **Warehouse** can handle multiple **Orders**. |
| **Order-Delivery Driver** | Order to Delivery Driver | One-to-One or One-to-Many | An **Order** is assigned to a single **Delivery Driver**, but one **Delivery Driver** can manage multiple **Orders**. |
| **Warehouse-Product** | Warehouse to Product | One-to-Many | A **Warehouse** stores many **Products**, but each **Product** belongs to one **Warehouse**. |

A light bulb with lines and dots

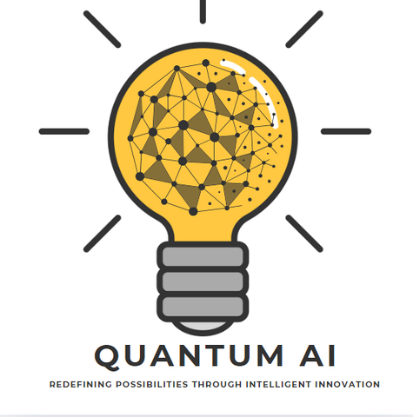
AI-generated content may be incorrect.Figure 2 ( Different types of Entity Realtionships)



### **Illustrative Data Flow Example**

Let’s go through a practical scenario involving the relationships discussed *(Databasestar.com,2023)*:

1. **Customer Places an Order (Customer-Order Relationship)**:
   1. **Customer A** logs onto the Speedy Gro platform and places an order for **Product A** and **Product B**.
   2. This action creates an **Order** entry, linking **Customer A** via their **Customer\_ID** to the **Order**.
2. **Order Fulfilled by Warehouse (Order-Warehouse Relationship)**:
   1. **Order 001** is assigned to **Warehouse 1**, which has **Product A** and **Product B** in stock.
   2. The **Warehouse** processes the order, decreasing the available stock of these products.
3. **Delivery Driver Assigned (Order-Delivery Driver Relationship)**:
   1. **Order 001** is assigned to **Driver A**, who will deliver the products.
   2. Depending on the business, **Driver A** may deliver just **Order 001** (one-to-one) or multiple orders in the same trip (one-to-many).
4. **Warehouse Stores Products (Warehouse-Product Relationship)**:
   1. **Warehouse 1** stores **Product A**, **Product B**, and **Product C**.
   2. If **Order 001** needs **Product A**, it will be fulfilled from **Warehouse 1**.



#### **6. Logical Data Model**

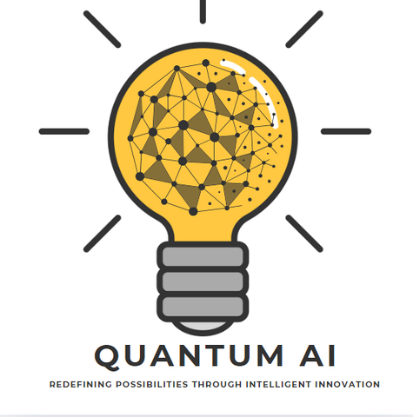
**Detailed Logical Data Model**

The detailed logical data model defines the data structures with precise relationships:

1. **Customer** (Customer\_ID, Name, Address, Phone, Email, Payment\_Method)
2. **Delivery Driver** (Driver\_ID, Name, Vehicle\_ID, Phone, Shift)
3. **Warehouse** (Warehouse\_ID, Location, Capacity)
4. **Product** (Product\_ID, Product\_Name, Quantity\_Available, Price, Warehouse\_ID)
5. **Order** (Order\_ID, Customer\_ID, Warehouse\_ID, Order\_Date, Delivery\_Date, Delivery\_Status, Total\_Amount)
6. **Order\_Product** (Order\_ID, Product\_ID, Quantity\_Ordered)

**Data Flow Diagrams (DFDs)**

1. **Customer places an order**: The customer places an order that triggers data flow to the warehouse and delivery system.
2. **Warehouse processes the order**: The warehouse checks for product availability and updates inventory.
3. **Delivery driver assigned**: Once inventory is confirmed, a delivery driver is assigned to deliver the order.



***7. Our Testing Environment Demos the Data Design and Behavior***

Data Model Simulation Tool The data model got its first trials using Lucid chart as our platform. The simulation demonstrates how delivery data passes between customers vehicles warehouses during live operations. Interactive Simulation Process

1. The Customer sends their order request to the mobile app.

2. After processing the order the Warehouse updates the stock records.

3. The Delivery Driver receives orders based on their present location and readiness.

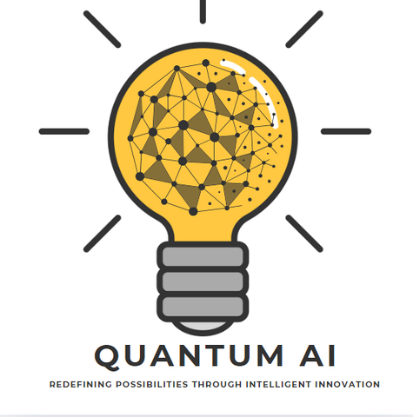
4. The Driver completes the delivery by sharing results to the system.

### A light bulb with lines and dots AI-generated content may be incorrect.

### **8. Understanding the ER Diagram**

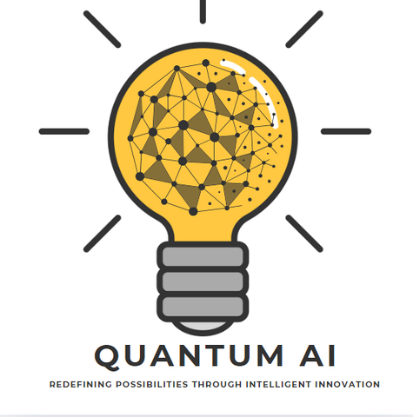
The ER diagram fills in the view of how different components in the Speedy Gro system interact (UC Davis). While important, the main entities in it are customers, orders, products, warehouses, and delivery drivers. Customers order various products kept in warehouses, and every ordered item is assigned to a particular delivery driver so that goods will reach the proper destination. It also presents, within the model, tracking mechanisms to enable Speedy Gro to track each order status, where their delivery drivers are, and what is in stock at the warehouses. The structure will assure smooth flow of data and facilitate operational efficiency. The ER diagram will have different entities and the relationship between entities.

The following are the details for the customer entity: customer ID, first name, surname, email, and phone number. Order entity: It stores the information of orders including order ID, customer ID, warehouse ID, assigned delivery driver ID, order status, order datetime, delivery fee, total amount, and expected delivery datetime. Product-Product ID, product name, price, in-stock, and warehouse ID are stored in the Product entity. Warehouse-related information such as warehouse ID, location, and address ID is kept within the Warehouse entity. The Delivery Driver entity includes driver-specific information like driver ID, first name, last name, contact information, vehicle details, and vehicle number. The



relationship among these entities defines how the system will work. A customer can place multiple orders, and each order is placed by a single customer, resulting in a one-to-many relationship.

Every Order may contain many Products, and every Product can be assigned to many Orders. That's an example of a Many-to-Many relationship, which is handled through the Order\_Product table. Every product is stored in one Warehouse. This is an example of a One-to-Many relationship: every warehouse may stock many products. Every Order is assigned to one Delivery Driver, but every driver may handle many orders - yet another example of a One-to-Many relationship. Address will be associated with the Warehouse, where every warehouse has only one address. This ER model will ensure that all critical data elements are well-structured and managed efficiently so as to enable Speedy Gro to maintain operational transparency and effectiveness.

A diagram of a product

AI-generated content may be incorrect.

Figure 3 (Entity Relationship Diagram for Speedy Gro)

### A light bulb with lines and dots AI-generated content may be incorrect.

### **9. Customer Management in the Data Model**

### Any organization functions through its customer base which requires their data to be arranged properly for easy retrieval. Our system assigns individual IDs for customers that can access information regarding their complete contact details including name and email and phone number. The system enables users to store several delivery addresses for varied distribution points. The system implements an organizational structure for data storage which enables convenient access to information at any time.

### **10. Order Processing and Management**

‘Order' part in our system reflects every purchase the customer has. When the customer places an order, it records the customer's unique ID assigned to that customer. An order

will have items in it containing the name of the product and price with their quantity. Products are shipped according to their availability from the warehouses. A driver is assigned so that the packages reach the customers without any damage. The status can be easily assessed whether the order is pending, packed, shipped, or delivered, both to customers and administrators.

A light bulb with lines and dots

AI-generated content may be incorrect.

**11. Real-Time Tracking of Delivery Drivers**

It will involve tracking the driver who will deliver the order to the customer for sure, faster, and quicker. A driver has a unique ID with all his personal information, such as name, contact, vehicle number, and the like. Immediately after the allocation of an order;

The driver's movement would be traced automatically in real-time, meaning a customer gets to receive notifications instantly regarding his/her order status. Besides that, Speedy Gro will be able to use GPS integration for route optimization of delivery, therefore minimizing delays.

The performance of the driver would be recorded in the system; thus, customer ratings could also be monitored by the company to improve service quality.

### **12. Inventory Management Across Warehouses**

Warehouses are the heart and soul of product availability at just about the right time. Each of the warehouses has a unique ID and is attached with an address in the system. Each warehouse location holds an assigned inventory for its linked products. Our system enables easy identification of product supplies to help us order restocking items on time.

A light bulb with lines and dots

AI-generated content may be incorrect.

When warehouse inventory levels decrease the system can notify the team to re-fill the supplies and enhance distribution processes.

**13. Relationship Between Orders and Products**

Here, there is an intermediary table, 'Order\_Product,' in the system that connects orders to products *(Geeks for Geeks)*. The system would extract quantitative data for items of the quantity level. The software design enables Speedy Gro to generate reports on both sales patterns and customer buying habits. The system permits orders only for products that remain in stock to avoid customer attempts at placing unavailable item orders. The data-driven method leads to improved customer shopping quality.

### **14.** **Order Status and Customer Feedback**

The status will be viewed by the customer upon completion of payment. It normally consists of four stages of order status: 'Pending,' 'Packed,' 'Out for Delivery,' and

A light bulb with lines and dots

AI-generated content may be incorrect.

'Delivered.' The first stage is that the orders are pending. When the staff packs those items, the status changes to 'Packed.' As a delivery driver collects from the warehouse the system automatically updates the status to 'Out for Delivery.' The final destination of the product occurs at the customer where the system registers an event as 'Delivered'. Users can track their item conveniently using smartphones as well as laptop devices without any difficulties. Speedy

Gro can evaluate the delivery service for future improvement through user feedback provided within the application and delivery system.

### **15. How the Application Works Internally**

Upon receiving an order via the Speedy Gro app, the system checks product availability in the nearest warehouse. An order will then be created and associated with a customer and a specific warehouse. Then it will assign a delivery driver according to location and availability while concurrently updating the order status in real time. The warehouse staff processes the order, packs the items, and hands them over to the assigned driver. From

A light bulb with lines and dots

AI-generated content may be incorrect.

order placement to final delivery, every stage of the process is notified to the customer. The app also gathers feedback that helps Speedy Gro in enhancing its operations.

### **16. Conclusion**

The data model designed for Speedy Gro effectively meets the main challenges of operation, such as order management, tracking of delivery, and monitoring of inventory. By structuring the data efficiently, the system sees to it that all the processes of the business function well.

Real-time tracking keeps customers updated on the status of their orders, while proper warehouse management guarantees product availability. This will enable The data model enables all companies to function efficiently on structured databases for handling both external and internal processes. Speedy Gro uses this data model to support current technological advancements that will underpin its future growth and expansion.

A light bulb with lines and dots

AI-generated content may be incorrect.

#### **17. Challenges and Considerations**

The management of orders along with inventory and deliveries depends on maintaining precise and current data information. An order tracking system together with warehouse inventory updates need to work in real time.

**Scalability and Flexibility**

The model needs to remain adaptable to expand capabilities along with increasing warehouse facilities and addition of more delivery personnel and enhanced customer order capacity.

**Data Security**

To keep customers safe and adhering to security laws, customer information, alongside delivery details and payment data, must be protected safely both in storage and during transmission.

#### **18. Conclusion**

Data Model Speedy Gro’s data models help the company work successfully by showing what information needs to be stored. The models help all parts of Speedy Gro work together well.

#### A light bulb with lines and dots AI-generated content may be incorrect.

#### **10. References**

* Lucidchart, (www.lucidchart.com)
* UC Davis, Conceptual Modeling using the Entity-Relationship Model, [2-er.pdf](https://web.cs.ucdavis.edu/~green/courses/ecs165a-w11/2-er.pdf)
* Databasestar.com,2023, A Guide to the Entity Relationship Diagram (ERD),

[A Guide to the Entity Relationship Diagram (ERD) - Database Star](https://www.databasestar.com/entity-relationship-diagram/)

* Geeks for Geeks, Introduction to ER Model,

[Introduction of ER Model - GeeksforGeeks](https://www.geeksforgeeks.org/introduction-of-er-model/)