

# Order Management System ER Diagram

The **ER Diagram for order management system** reveals the relationships of its entities to form the project database design. This describes the logical structure of the system's database or data storage. It is done by identifying the order process entities, their properties, and the interactions between them.

The **database design** is sketched out using **order management system project ER diagrams**. This database sketch becomes the actual basis of the system's data storage that will serve as data destination and source.

## What is an ER Diagram?

The ER Diagram is referred to as the **order management system database design**. This ER diagram is the graphical depiction of relationships between all the entities involved in the system. Its major components are Entities, Attributes, and Relationships.

In DBMS, the **order management system ER Diagram** is used to build and troubleshoot relational databases. It works best with DFD (Data Flow Diagram), which is responsible for data movement. Designing the order management system database would be much easier with the help of ER diagram.

The **order management system ER diagram** was based on an order management database design. The system's function is to encode customer information and transaction. Also, the order management admin must have access to the customer information for reports and inventory purposes. The data used in these transactions must be managed and secured well and the use of **ER Diagram for the Order Management System** is needed.

## Importance of ER Diagram

The **importance of ER diagram** for order management system is to help in modeling its data storage or database. It is the basis of the project's database foundation for construction. This entity-relationship diagram (ERD) also aids in defining the data types to be stored such as their attributes and characteristics.

In addition to that, the ER Diagram also describes how an entity interacts with other entities. All other real-world projects are presented with ER Diagrams (database designs). To display the details and attributes of a data store, the **er diagram for order management system** is used in conjunction with its data flow diagram. They are very important in building a relational database because they let us visualize how data is connected generically.

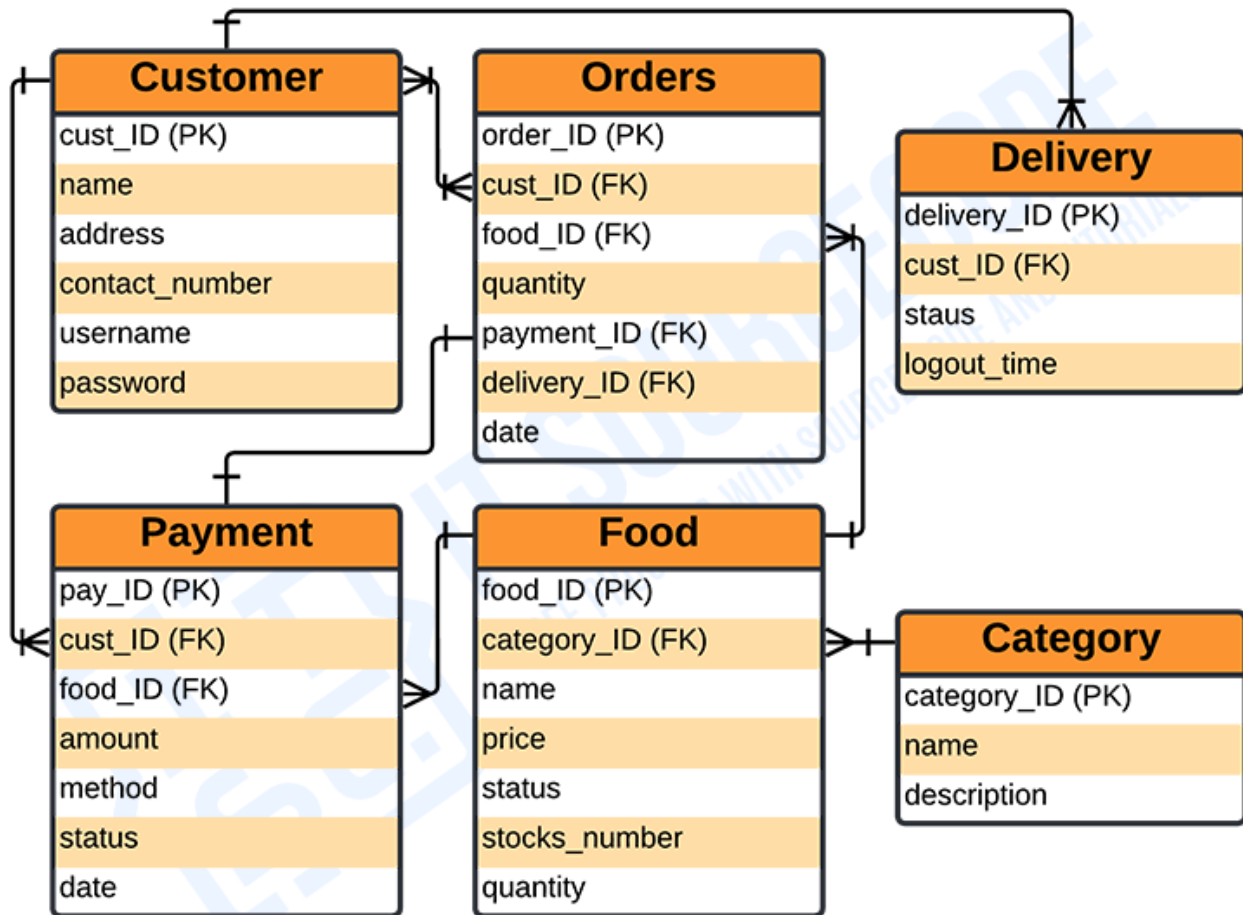
Entity-relationship diagrams are utilized in software engineering during the planning phase of software development. It aids in the identification of various system constituents and their

interrelationships. **Order Management System ER Diagram** is also used as the foundation of its DFD (Dataflow Diagram)

The **database design for Order Management System** is the same as its **ERD**. They were used to emphasize the **order management system database design** for data storage to determine the data to be stored and their attributes. This is to inform developers about the system's structure in terms of data storing.

## Entity-Relationship (ER) Diagram for Order Management System

**ER Diagram of Order Management System** shows the included entities (data) and their supposed functions (attributes). Each of them was represented by a table to illustrate their characteristics and relationships with each other.



*ER Diagram for Order Management System*

Based on the illustration given, you'll be able to know the whole purpose of the database design of **ER Diagram for Order Management System**. That is to show the presence of a database to be used as the storage of all the data that enters the system and as the source of all the

output that the users require That is how important the Database or the **ER Diagram design for the Order Management System** function well.

## Order Management System ER Diagram Tables

These tables below provide the complete database table details such as **Field Name, Descriptions, data types, and character lengths**. Each of these tables represents the characteristics and the attributes of data storage.

The **field** column presents the names of each database's attributes, the **description** column gives the complete thought of each attribute, the **type** column is their data type and the **length** is for their character lengths.

### Table Name: Customer

Field	Description	Type	Length
<b>cust_ID (PK)</b>	Customer ID	Int	11
<b>name</b>	Customer Name	Varchar	255
<b>address</b>	Customer Address	Text	255
<b>contact_number</b>	Contact Number	Varchar	255
<b>username</b>	Username	Varchar	255
<b>password</b>	Password	Varchar	255

### Table Name: Food

Field	Description	Type	Length
<b>food_ID (PK)</b>	Food ID	Int	11
<b>category_ID (FK)</b>	Category ID	Varchar	255
<b>name</b>	Food Name	Varchar	255
<b>price</b>	Food Price	Text	
<b>status</b>	Food Status	Varchar	255
<b>stocks_number</b>	Stocks Number	Varchar	255
<b>quantity</b>	Quantity	Int	11

### Table Name: Category

Field	Description	Type	Length
<b>category_ID (PK)</b>	Category ID	Int	11
<b>name</b>	Category Name	Varchar	255
<b>description</b>	Description	Text	

### Table Name: Payment

Field	Description	Type	Length
<b>pay_ID (PK)</b>	Payment ID	Int	11
<b>cust_ID (FK)</b>	Customer ID	Int	11
<b>food_ID (FK)</b>	Food ID	Int	11
<b>amount</b>	Amount	Varchar	255

<b>method</b>	Payment Method	Varchar	255
<b>status</b>	Payment Status	Varchar	255
<b>date</b>	Date of Payment	Date	

### Table Name: Orders

Field	Description	Type	Length
<b>order_ID (PK)</b>	Order ID	Int	11
<b>cust_ID (FK)</b>	Customer ID	Int	11
<b>food_ID (FK)</b>	Food ID	Int	11
<b>quantity</b>	Quantity Orders	Int	11
<b>payment_ID (FK)</b>	Payment ID	Int	11
<b>delivery_ID (FK)</b>	Delivery ID	Int	11
<b>date</b>	Date of Order	Date	

### Table Name: Delivery

Field	Description	Type	Length
<b>delivery_ID (PK)</b>	Employee Id	Int	11
<b>cust_ID (FK)</b>	First Name	Varchar	30
<b>status</b>	Last Name	Varchar	30

The tables given will be the basis for developers on how would they would work on the **order management system database design**. It has the complete description of the database and will be applied to the program as data storage the same as the names given to each of the tables.