<u>“INVENTORY MANAGEMENT SYSTEM”</u>

Submitted as **mini project report** for
CS251-Database Systems

Submitted by:

Dev Agrawal (201ME169)

Kishan Singh (201EE269)

Dhruv Yadav (201ME318)

INTRODUCTION

In a world where data is important to thrive in the present ecosystem of a rapidly rising consumer base, any business, large or little, requires a system to properly store their data that will help them adapt to their consumers' demands. Databases such as Microsoft SQL Server, Oracle, My SQL, and PostgreSQL are used to persist data and store it effectively, allowing the system to scale readily as demand grows.

Managing and maintaining inventory is one of the most challenging responsibilities for organizations, since it allows them to make timely decisions regarding stocking up on items and ensuring that transactions run smoothly. Most small firms retain a register for this purpose, however this results in inefficiencies, which can be detrimental to the company, therefore, using our database system would help save paper and provide digital worldwide accessibility to the consumer in addition to many other advantages.

We want to target individual owners who can use databases to see available space at any given time in the future and accept storage contracts well in advance, maximizing capacity use and profit.

This is the problem we are aiming to solve through our project.

ELEMENTS OF SOLUTION

The project's goal is to create a database that will assist small business models that cannot afford a full-fledged bloated enterprise software in managing inventory by storing data about available stocks in a MySQL database, allowing the company to make timely supply decisions.

There are numerous important components to an inventory management system. Inventory control is based on tracking two main functions of a warehouse: receiving and shipping, as well as other activities such as inventory movement or relocation. The value of raw materials is decreased, while the value of finished goods is increased.

The model is implemented using **7** tables:-

1. **Product table:** Consisting the data regarding all the products that is the product_id,Name,Price, and the on hand inventory.

2. **Supplying:** This table was created by decomposing the suppliers table to meet the constraints of BCNF. It consists of supplier id, product_id and quantity.

3. **Supplier:** Contains data of all the suppliers including the supplier_id and the name.

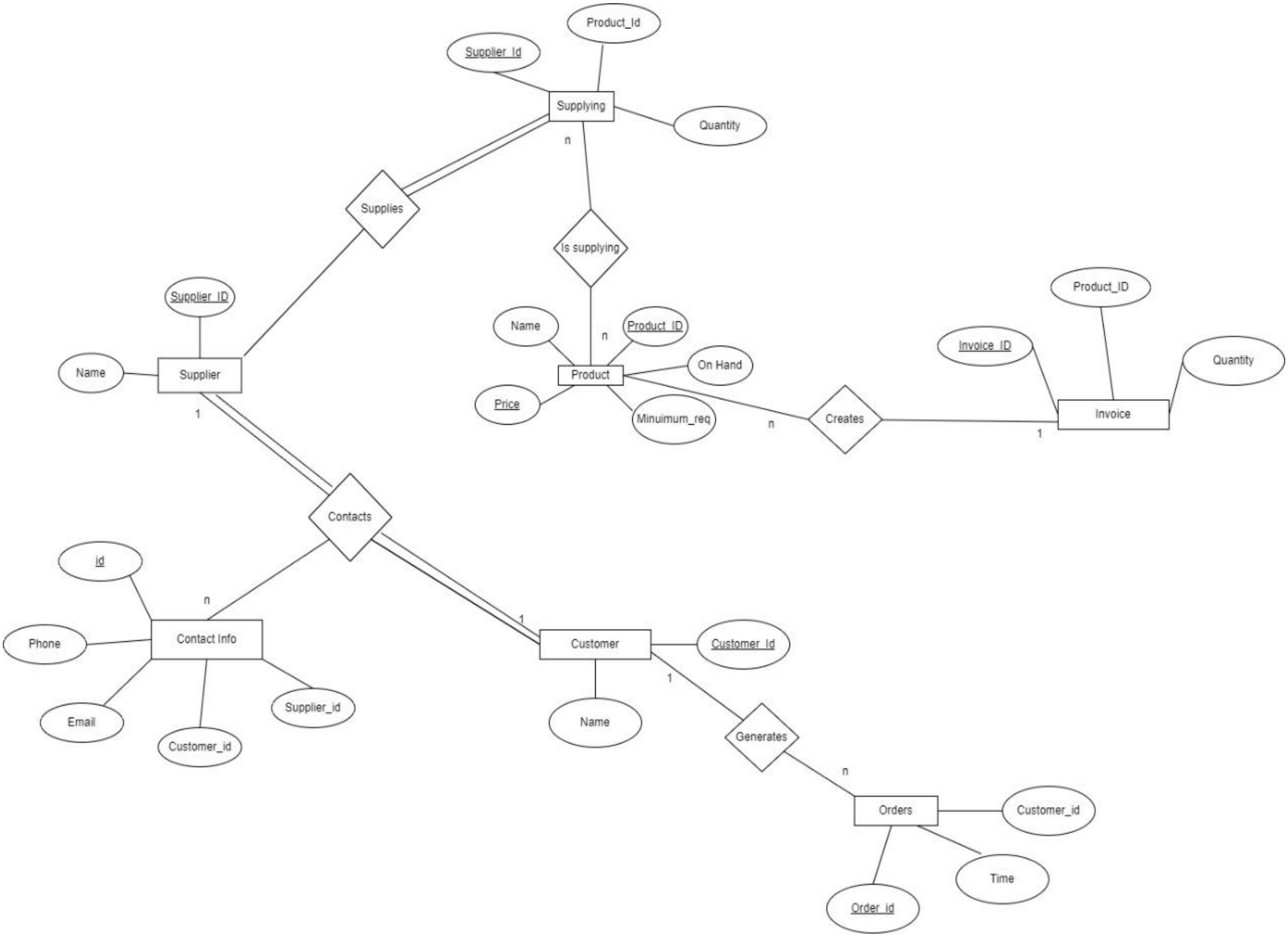
4. **Customer:** holds all of the customer's information, such as their customer id and name.

5. **Contact Info:** contains the contact info of the customer and the supplier the attributes consist of supplier_id, customer_id, phone no., email id and id.

6. **Invoice:** contains the invoice of all the orders placed by the customers in the database. It contains the invoice_id, product_id, quantity

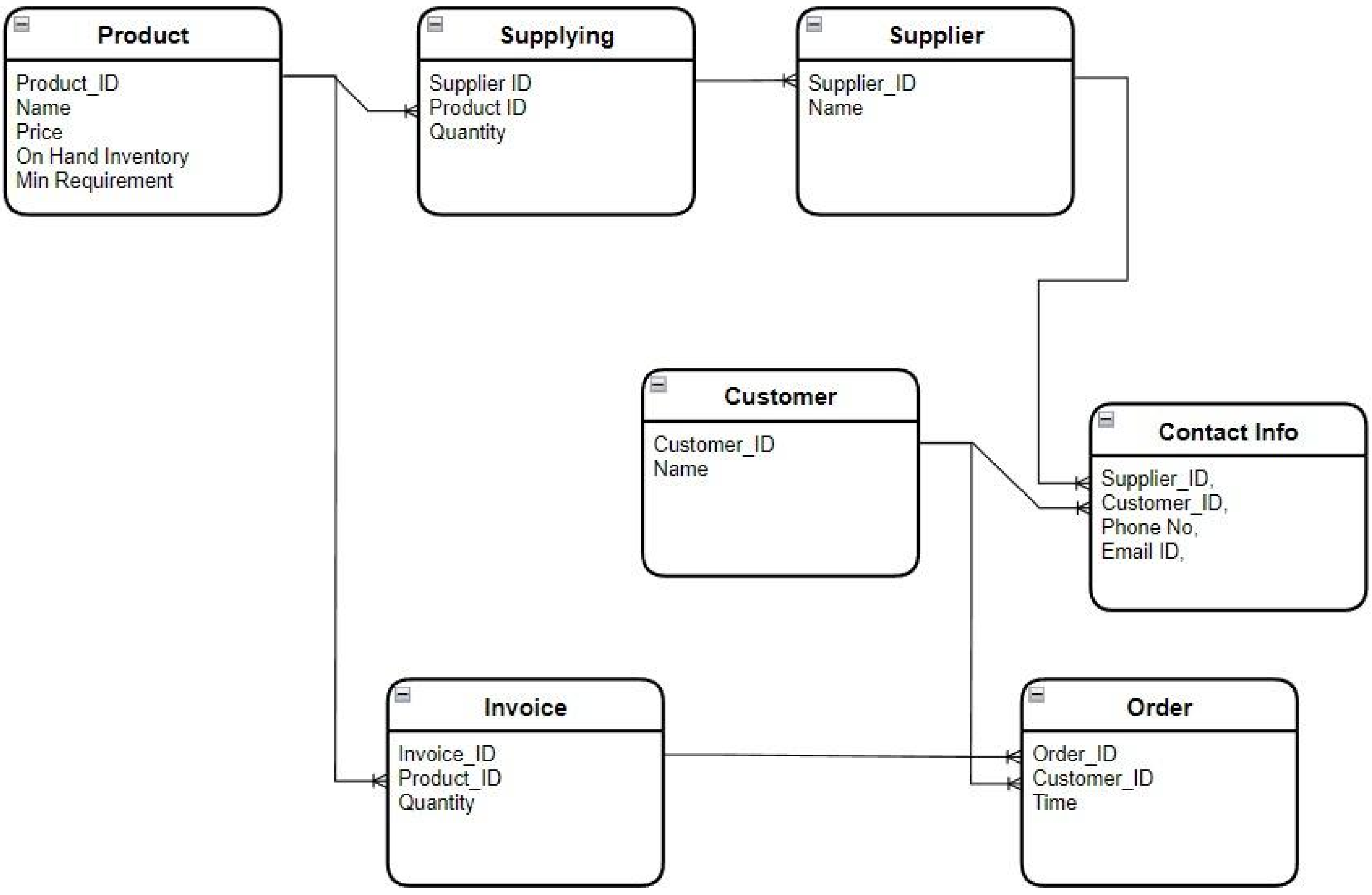
7. **Order:** it contains the info of all the orders in the database. It has the order_id, customer_id and time.

ER DIAGRAM


Product_Id
Supplier_Id
Supplying
Quantity
n
Supplies
Is supplying
Supplier_ID
Name
Name
Product_ID
n
On Hand
Supplier
Product
Price
Minuimum_req
1
Product_ID
Invoice_ID
Quantity
Invoice
n
Creates
1
Contacts
id
n
Contact Info
Phone
Email
Customer_id
Supplier_id
1
Customer
Customer_Id
Name
1
Generates
n
Orders
Customer_id
Time
Order_id

RELATIONAL SCHEMA

Inventory Management System



SQL QUERIES TO CREATE THE TABLES

```
CREATE TABLE "Product" (
	"Product_ID"	INTEGER NOT NULL,
	"Name"	TINYTEXT NOT NULL,
	"Price"	INTEGER NOT NULL,
	"On Hand Inventory"	INTEGER NOT NULL,
	"Min Requirement"	INTEGER NOT NULL,
	PRIMARY KEY("Product_ID" AUTOINCREMENT)
);

CREATE TABLE "Supplying" (
	"Supplier ID"	INTEGER NOT NULL,
	"Product_ID"	INTEGER NOT NULL,
	"Quantity"	INTEGER NOT NULL,
	FOREIGN KEY("Product_ID") REFERENCES
	"Product"("Product_ID"),
	PRIMARY KEY("Supplier ID")
);

CREATE TABLE "Supplier" (
	"Supplier_ID"	INTEGER NOT NULL,
	"Name"	TINYTEXT NOT NULL,
	FOREIGN KEY("Supplier_ID") REFERENCES
	"Supplying"("Supplier_ID")
	PRIMARY KEY("Supplier ID")
);

CREATE TABLE "Customer" (
	"Customer_ID"	INTEGER NOT NULL,
	"Name"	TINYTEXT NOT NULL,
	PRIMARY KEY("Customer_ID")
);
```

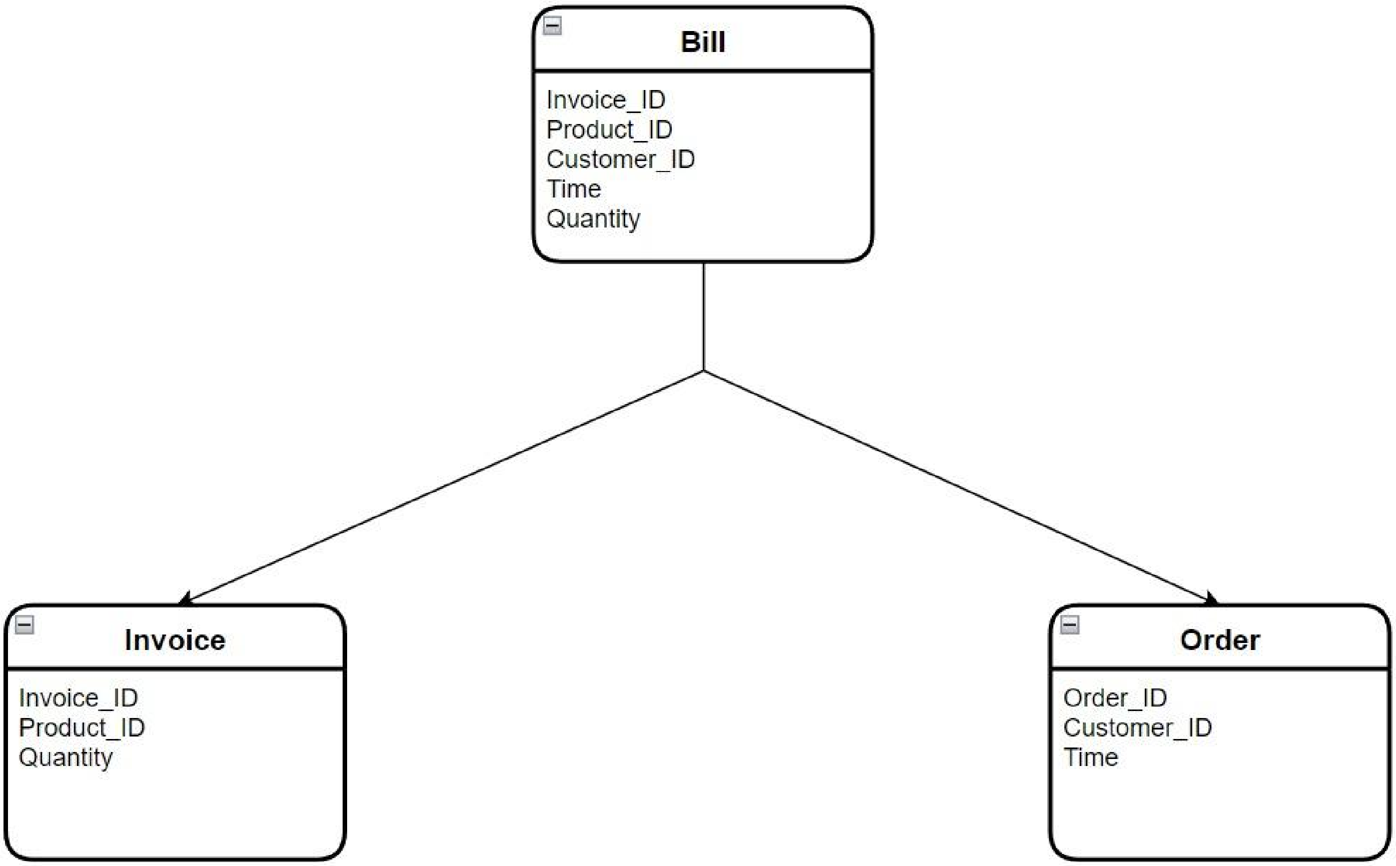
```
CREATE TABLE "Contact Info" (
        "Supplier_ID"           INTEGER,
        "Customer_ID"           INTEGER,
        "Phone No"              INTEGER UNIQUE,
        "Email ID"      TINYTEXT UNIQUE,
        "id"    INTEGER NOT NULL,
        PRIMARY KEY("id" AUTOINCREMENT),
        FOREIGN KEY("Customer_ID") REFERENCES
        "Customer"("Customer_ID"),
        FOREIGN KEY("Supplier_ID") REFERENCES
        "Supplier"("Supplier_ID")
);

CREATE TABLE "Invoice" (
        "Invoice_ID"            INTEGER,
        "Product_ID"            INTEGER NOT NULL,
        "Quantity"      INTEGER NOT NULL,
        PRIMARY KEY("Invoice_ID"),
        FOREIGN KEY("Product_ID") REFERENCES
        "Product"("Product_ID")
);

CREATE TABLE "Order" (
        "Order_ID"      INTEGER,
        "Customer_ID"           INTEGER NOT NULL,
        "Time"          smalldatetime,
        FOREIGN KEY("Order_ID") REFERENCES
        "Invoice"("Invoice_ID"),
        FOREIGN KEY("Customer_ID") REFERENCES
        "Customer"("Customer_ID"),
        PRIMARY KEY("Order_ID")
);
```

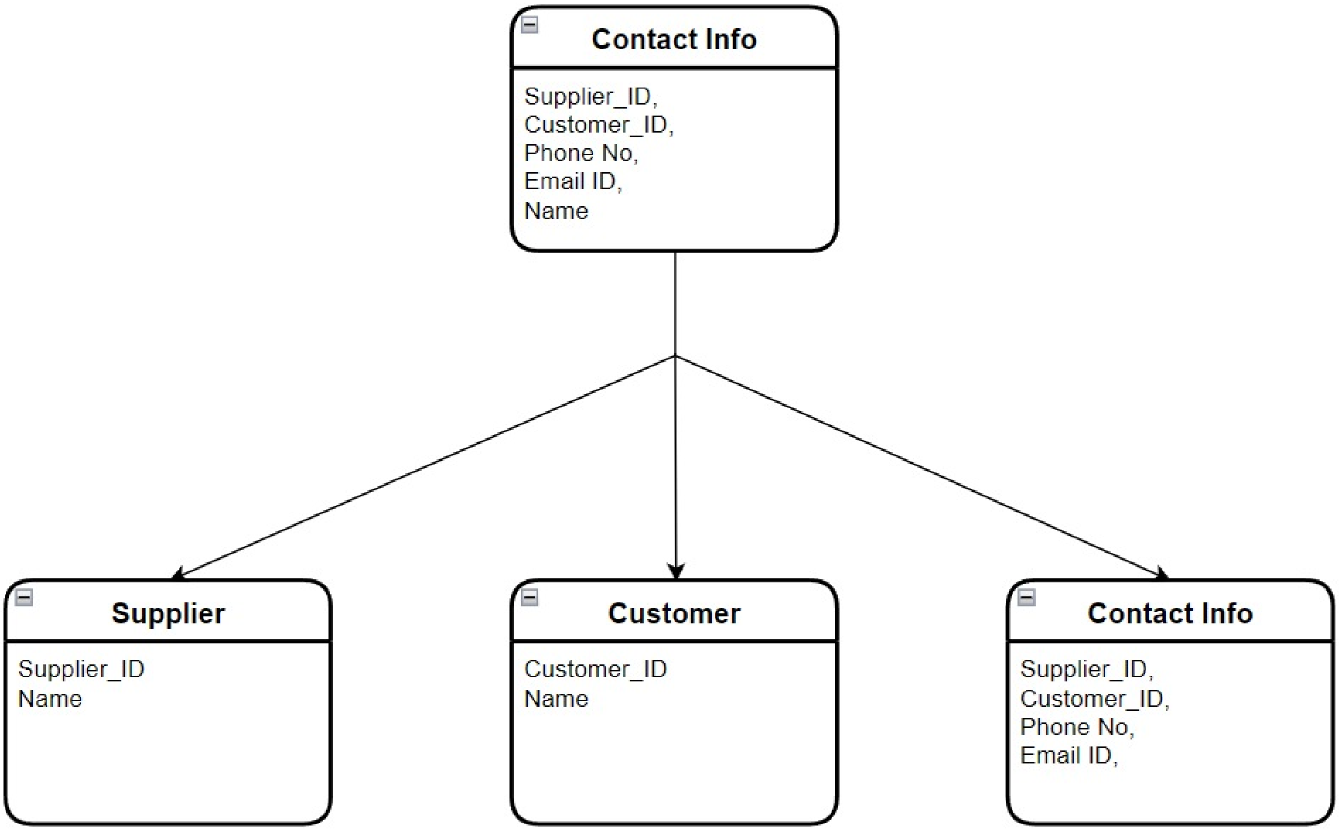
BCNF DECOMPOSITIONS OF RELATIONS

1)



Quantity and Time together may determine other attributes in some small scale datasets; thus to eliminate this possibility, the relation bill has been decomposed in order to normalize the relation.

2)



Here {Phone No, Email ID} is the candidate key, thereby making Phone No and Email ID as the Prime Attributes.

Customer_ID and Supplier_ID together can determine name, which is a violation of 3NF as a non prime attribute is determining another non prime attribute, portraying transitive dependency. Thus to normalize it and eliminate the transitive functional dependency, the relation was decomposed.

RESULT

Finally, we have a robust lightweight inventory management system that is easy to use and scalable for warehouse owners and anybody else interested in inventory management in any area of the world. We strive to make inventory management as efficient as possible, whether you're a store, library, or a professional corporation.

- The database has low data redundancy which makes searching and other queries time efficient.
- The database has a very broad user base thereby increasing the feasibility.
- The digital nature of the database makes it better than the trivial methods of maintaining inventory data and it also provides Digital world wide accessibility
- Devices such as Bar Code Scanners can be easily integrated with the database to provide the Product ID thereby making the program versatile, and adaptive to the user's needs