ER Diagram Analysis Report

1. Introduction

ER diagram is an entity-relationship model of a business system that incorporates customers, orders, invoices, payments, products, suppliers, events, tickets, and users. It is an examination of the inter-entity relations and cardinalities.

2. Entities and Their Relationships

Each entity in the database is defined with specific attributes and is linked to other entities through relationships.

2.1 Customers and Orders (1:M)

- **Entity: Customers** (customer_id)
- **Entity: Orders** (order_id, customer_id)
- **Relationship: One-to-Many (1:M)**

1. One **customer** can place **many orders**, but each **order** belongs to only one **customer**.

2.2 Orders and Order_Items (1:M)

- **Entity: Orders** (order_id)
- **Entity: Order_Items** (order_item_id, order_id, product_id)
- **Relationship: One-to-Many (1:M)**

2. One **order** can contain **multiple order items**, but each **order item** belongs to only one **order**.

2.3 Order Items and Products (M:1)

- **Entity: Order Items** (order_item_id, product_id)
- **Entity: Products** (product_id)
- **Relationship: Many-to-One (M:1)**

3. Many **order items** can reference the **same product**, but each **order item** is linked to one **product**.

2.4 Orders and Invoices (1:1)

- **Entity: Orders** (order_id)
- **Entity: Invoices** (invoice_id, order_id)
- **Relationship: One-to-One (1:1)**

4. Each **order** generates **one invoice**, and each **invoice** corresponds to exactly **one order**.

2.5 Invoices and Payments (1:1)

- **Entity: Invoices** (invoice_id)
- **Entity: Payments** (payment_id, invoice_id)
- **Relationship: One-to-One (1:1)**

5. Each **invoice** has exactly **one payment**, and each **payment** is linked to one **invoice**.

2.6 Products and Suppliers (M:1)

- **Entity: Products** (product_id, supplier_id)
- **Entity: Suppliers** (supplier_id)
- **Relationship: Many-to-One (M:1)**

6. Many **products** can be supplied by the **same supplier**, but each **product** has only **one supplier**.

2.7 Customers and Tickets (1:M)

- **Entity: Customers** (customer_id)
- **Entity: Tickets** (ticket_id, customer_id, event_id)
- **Relationship: One-to-Many (1:M)**

7. A **customer** can purchase **multiple tickets**, but each **ticket** is assigned to one **customer**.

2.8 Events and Tickets (1:M)

- **Entity: Events** (event_id)
- **Entity: Tickets** (ticket_id, event_id)
- **Relationship: One-to-Many (1:M)**

8. One **event** can have **many tickets**, but each **ticket** is assigned to one **event**.

2.9 Users and Roles (1:M)

- **Entity: Users** (user_id, role)
- **Relationship: One-to-Many (1:M)**

9. Each **user** has a **role** (Admin, Staff, or Customer), and multiple users can belong to the same **role**.

Date
Page

SQL
Inventory management system

Important Entities :-

01. Supplies (which related As Supplies_ID → product_ID)
02. product (which related product_ID → orders Item)
03. customess (which realated customes_ID → orders)
04. ordess (which related orders ID → orders_item → Invoice)

which, involves the details of Tickets, Event payment which related to invoice and students fee payment details)

05. Orders_Item (which involues the table like orders ID, product ID)
06. Invoices (which involves the Invoices_ID, $ payment)
07. payments (which makes Relation between payment ID and Invoice ID)
08. Usess (which involves the uses ID and Admin, staff, customes and uses)

Future Involvement

09. Events (which makes Relation between Event: Event_ID, Ticket, pagment, invoices may be with asess, orders ID, product ID, product)

10) Tickets (Ticket_ID, which has Relation with customer_ID, order ID, payment, Events, Invoice

11) Student (student ID → Events, payment, Ticket, invoices)

Supplies includes like ticket provides Event organizes and etc.,

Future expansion

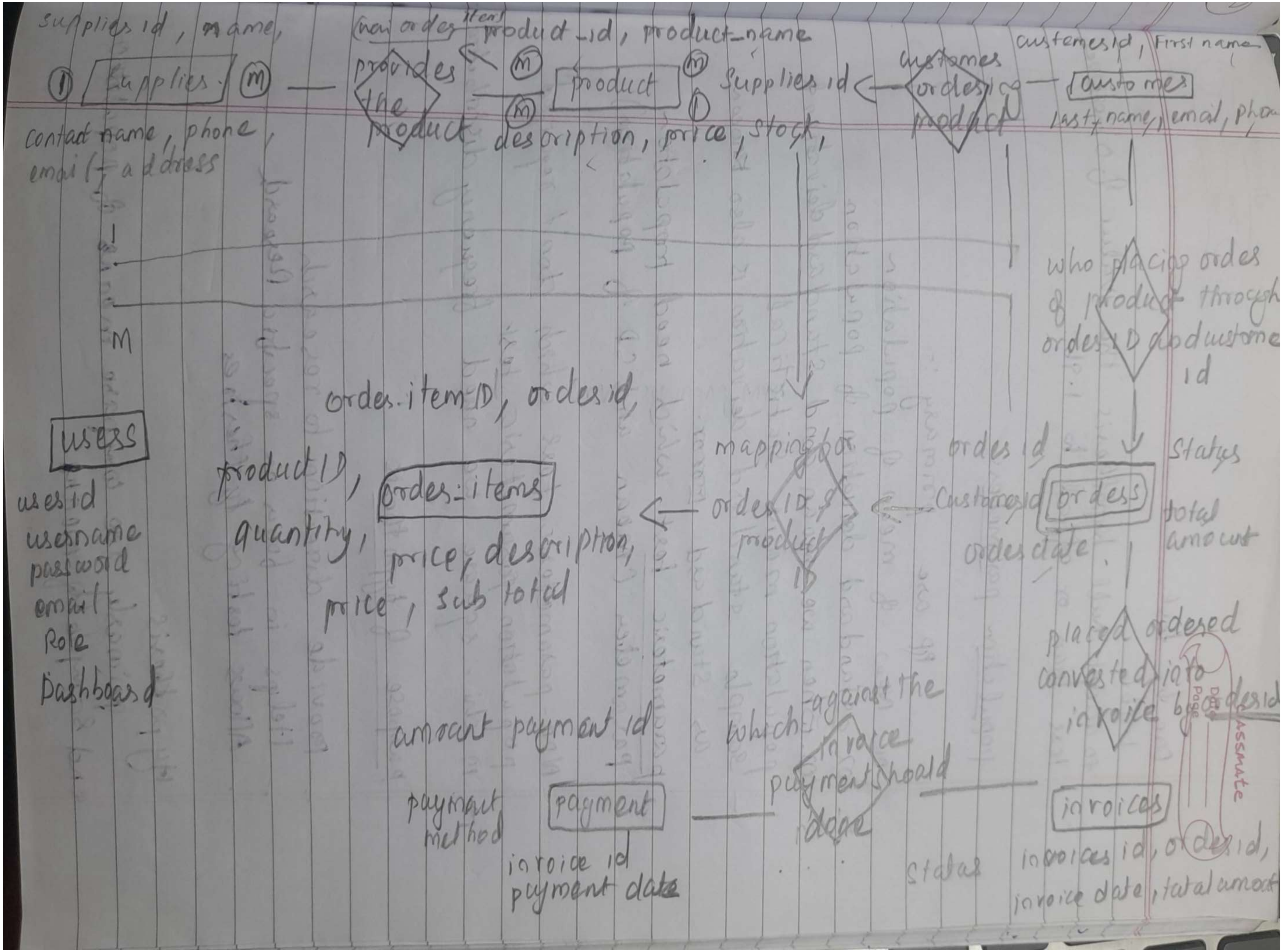
customer — who schedule Event based on previous Record — Events

Event id, customer id,

Invoice id, Event name, Event date, location, payment-id, descriptio

Event ticket management / sales

Event id, Customer id, ticket, ticket price, ticket id, payme, Seat mapping


supplies id, name,
Supplies
contact name, phone,
email, address
provides the product
nav order items
product_id, product_name
product
description, price, stock,
Supplies id
customer ordering product
customerid, First name
customers
last name, email, phone
who placing order of product through order ID and customer id
M
order.item ID, order id,
USERS
userid
username
password
email
Role
Dashboard
product ID,
orders_items
quantity, price, description,
price, sub total
mapping for order ID & product ID
order id
Customer id
orders
order date
Status
total amount
placed ordered converted into invoice by order id
amount payment id
payment method
payment
invoice id
payment date
which invoice payment should done against the
status
invoices
invoices id, order id,
invoice date, total amount

3. Summary of Relationships

Entity 1	Entity 2	Relationship Type
Customers	Orders	**1:M** (One customer can have many orders)
Orders	Order Items	**1:M** (One order contains multiple order items)
Order Items	Products	**M:1** (Many order items refer to one product)
Orders	Invoices	**1:1** (Each order has exactly one invoice)
Invoices	Payments	**1:1** (Each invoice has exactly one payment)
Products	Suppliers	**M:1** (Many products are supplied by one supplier)
Customers	Tickets	**1:M** (One customer can buy multiple tickets)
Events	Tickets	**1:M** (One event has multiple tickets)
Users	Roles	**1:M** (Multiple users can belong to the same role)

4. Conclusion

ER diagram is a great database schema for orders, payments, customers, suppliers, and events. The 1:M (One-to-Many) relationships support proper transaction handling, and the 1:1 (One-to-One) relationships between invoices and orders/payments ensure accounting record consistency.