

CSA0593

DBMS

ASSIGNMENT-1

K. Prathiksha

192321173

1. Designing a Dynamic E-commerce Database supporting products, categories, customers, orders, reviews

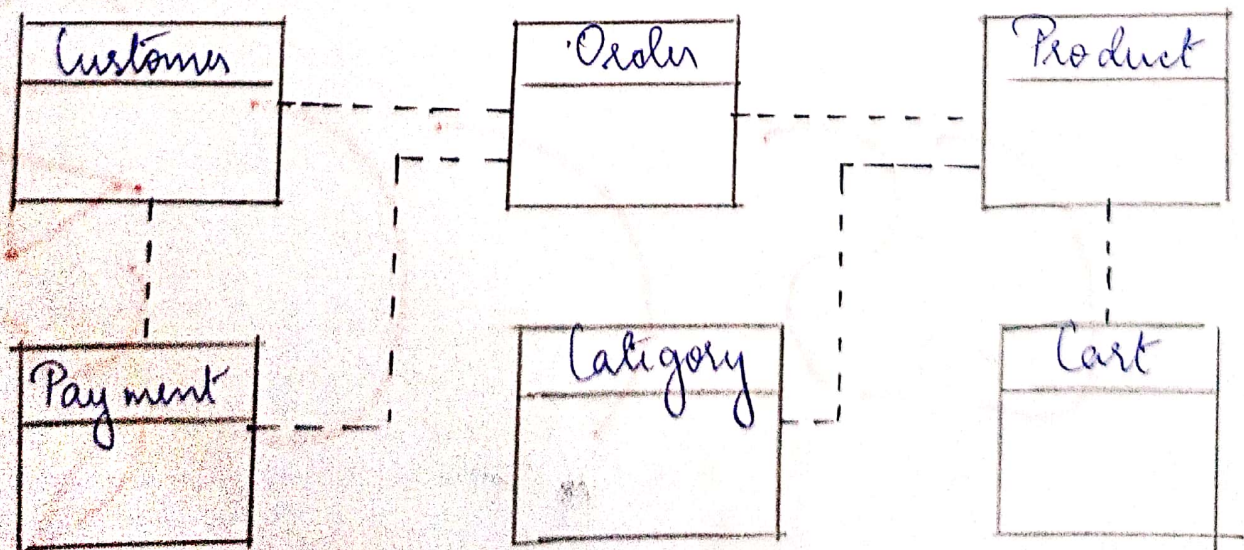
- * Model the relationships b/w products & categories ensuring efficient categorisation & retrieval

- * Implement stored procedures for product availability, cart updates and order processing

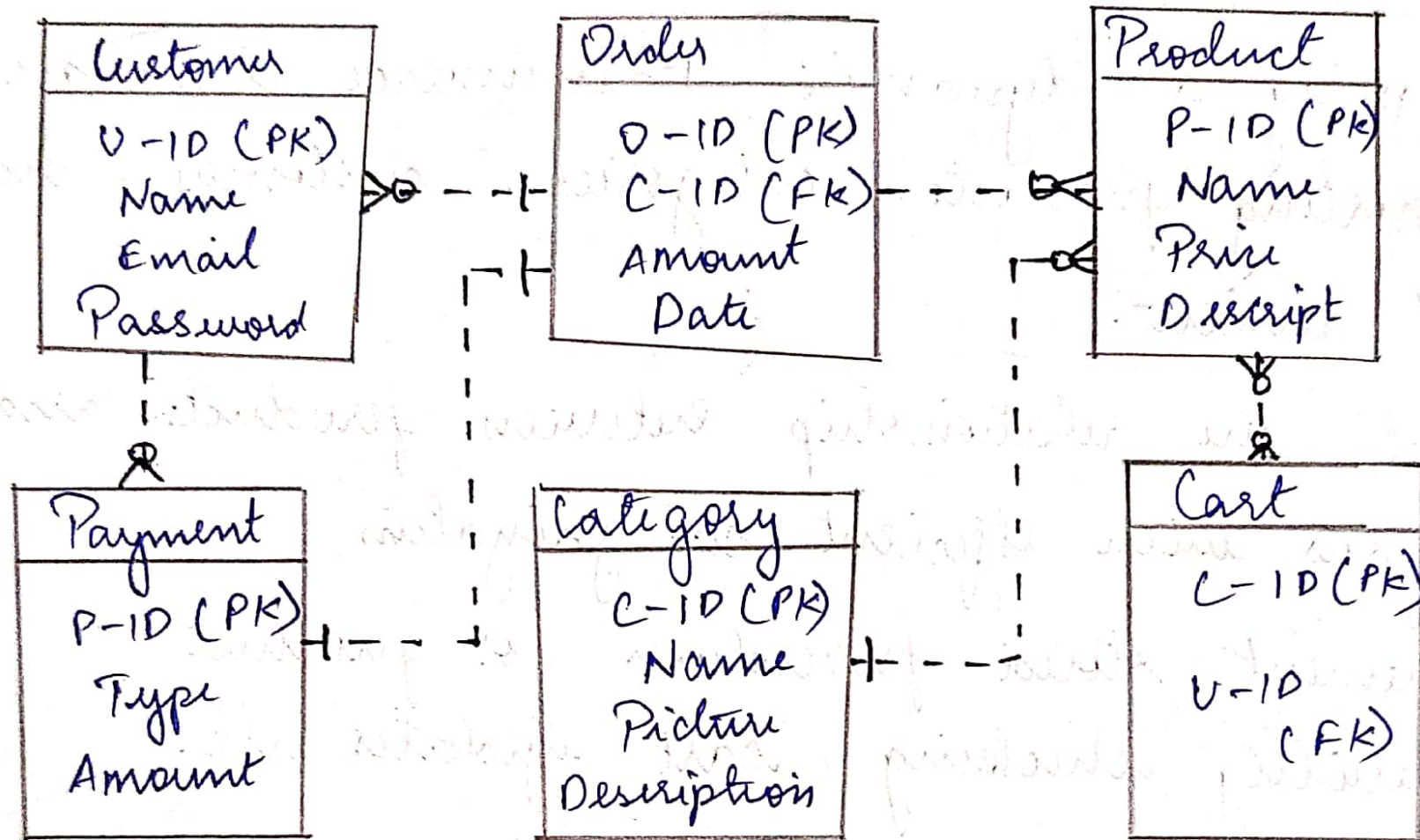
- * Add triggers to handle inventory updates, order cancellations, back-in-stock items

- * Normalise to 3NF, document each step of the design process and demonstrate working schema.

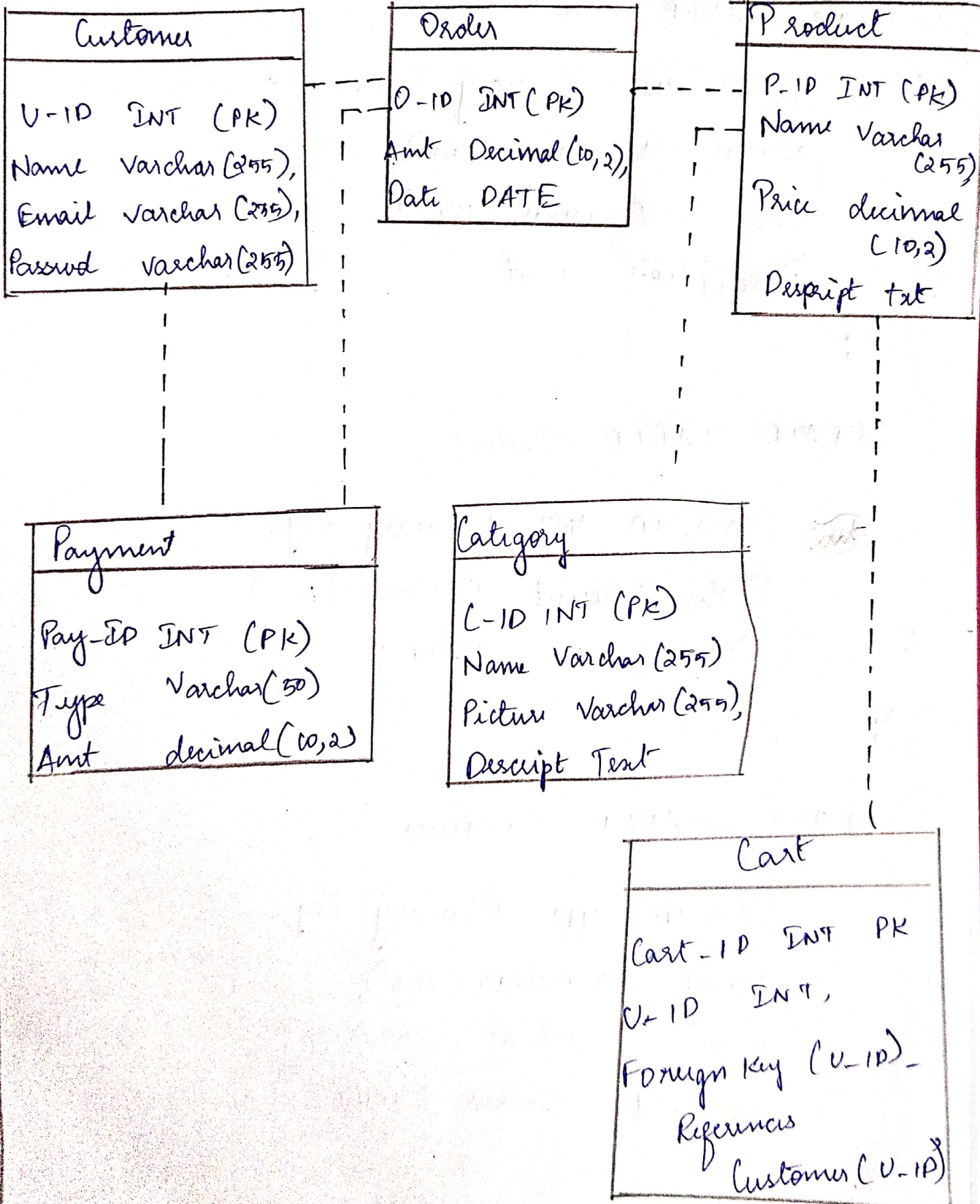
* Conceptual ER Model



* Logical ER Model



* Physical ER Model



* Creating tables

Create TABLE Product (

P_ID INT Primary key,

Name Varchar (255),

Price Decimal (10, 2),

Description Text

);

CREATE TABLE Order (

Order_ID INT Primary key,

Order-Amount Decimal (10, 2)

Order-Date DATE

);

CREATE TABLE Customer (

User-ID INT Primary key,

Name Varchar (255),

Email Varchar (255)

Password Varchar (255).

);

CREATE TABLE Payment (

Payment-ID INT Primary key,

Type Varchar(50)

Amount Decimal(10,2)

);

CREATE TABLE Cart (

Cart-ID INT Primary key

User-ID INT,

Foreign key (User-ID) references Customers (User-ID)

);

CREATE TABLE Category (

C-ID INT Primary key

Name Varchar(255)

Picture Varchar(255),

Description Text

);

checking product availability

```
CREATE PROCEDURE checkProductAvail (IN Prod INT)
BEGIN
    SELECT stock, is_available
    FROM Products
    WHERE prod P_ID = Prod ID;
END //
```

Triggers

Inventory Updates

```
DELIMITER //
```

```
CREATE TRIGGER UpdateInventory
AFTER INSERT ON Orders
FOR EACH ROW
BEGIN
    UPDATE Products SET stock = stock - New.quantity
    WHERE Prod_ID = New.Prod-ID ;
    IF (SELECT stock FROM Products WHERE
        P_ID = N.prod-ID) <= 0 THEN
        UPDATE Products Set is_available = FALSE
        WHERE P_ID = New.P_ID; END IF;
END //
```

Order Cancellations

```
CREATE TRIGGER RestoreInventoryCancel  
AFTER UPDATE ON Orders  
CREATE  
FOR EACH ROW
```

```
BEGIN
```

```
IF New.status = 'Cancelled' THEN
```

```
UPDATE Products P
```

```
JOIN Order-Items ON oi.prod-ID = p.prod-ID
```

```
SET p.stock = p.stock + oi.quantity
```

```
WHERE oi.order-id = New.order-ID;
```

```
END IF;
```

```
END //
```

Back-in-stock Notification

```
DELIMITER //
```

```
CREATE TRIGGER NotifyBackInStock
```

```
AFTER update ON Products
```

```
FOR EACH ROW
```

```
BEGIN
```

```
IF New.stock > 0 and New.is-available = FALSE,
```

```
UPDATE Products SET is-available = TRUE
```


WHERE prod_ID = New-Prod-ID;

END IF;

END//

Conclusion

The creation of an entity relationship model for an E-commerce site entails a lot of processes that involve evaluating different sides like entity identification, relationship definition, normalization, data integrity etc. A database designed with precision underpin the efficacy, reliability and scalability of a convenient E-commerce platform in order to attract more customers.