

# **EXOT ANIMAL E-COMMERCE**

**Lilly Parham, Pamela Alvarado-Zarate & Gracie Rehberg**  
**MSA 8040 - Data Management**

# PROJECT TASK

## E-Commerce Platform with Customer and Sales Analytics

### Description

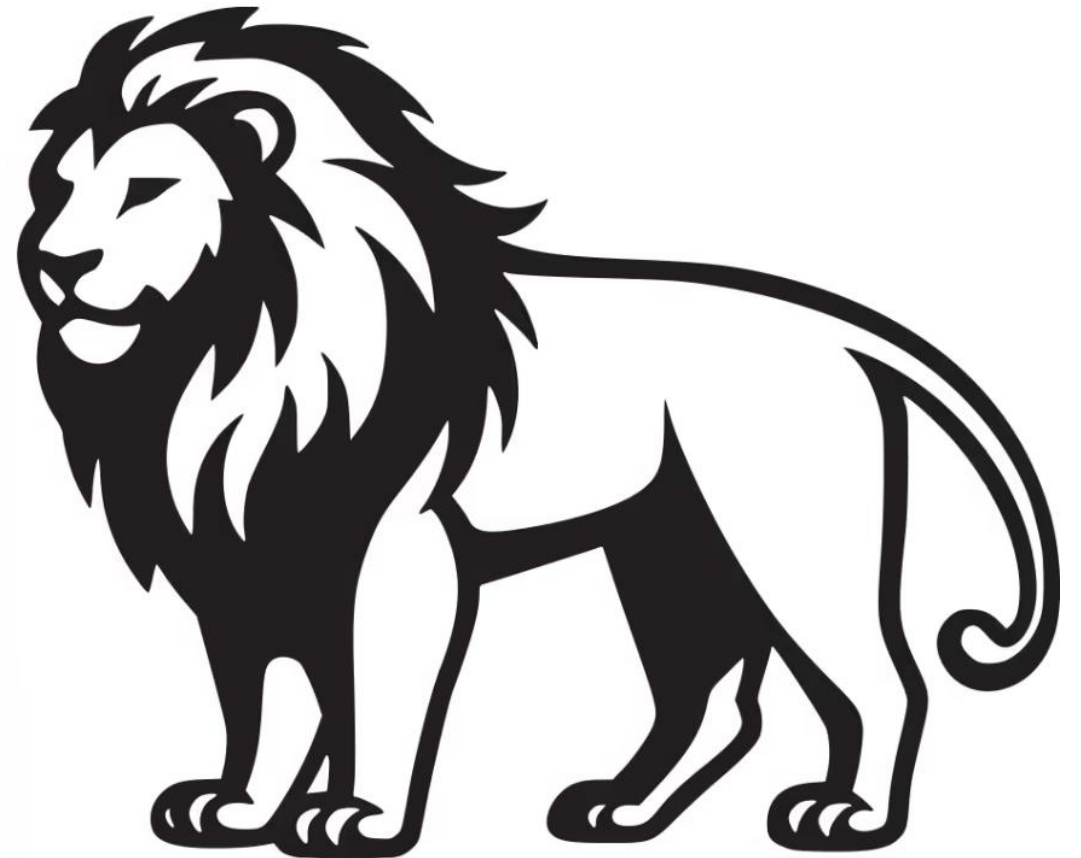
- Create an e-commerce database products, categories, users, orders, reviews, payment details.
- Implement functionalities like product browsing, order processing, and customer reviews.

### Challenge

- Add a sales analysis dashboard to show trends in customer purchases, most popular products, seasonal variations, and profit per order.
- Implement triggers and stored procedures to handle updates efficiently.

# EXOT ANIMAL E-COMMERCE

A platform dedicated to delivering **exotic wildlife and their necessities**, ensuring safety and care standards are upheld to safely transition them to their new home.



# THE DATABASE

## 10 Entities    11 Relationships

**Follows 3<sup>rd</sup> Normal Form (3NF)**

---

**ALL** tables with...

...unique, primary &  
foreign keys

...required domains

**User:** ID,  
name, email,  
and address

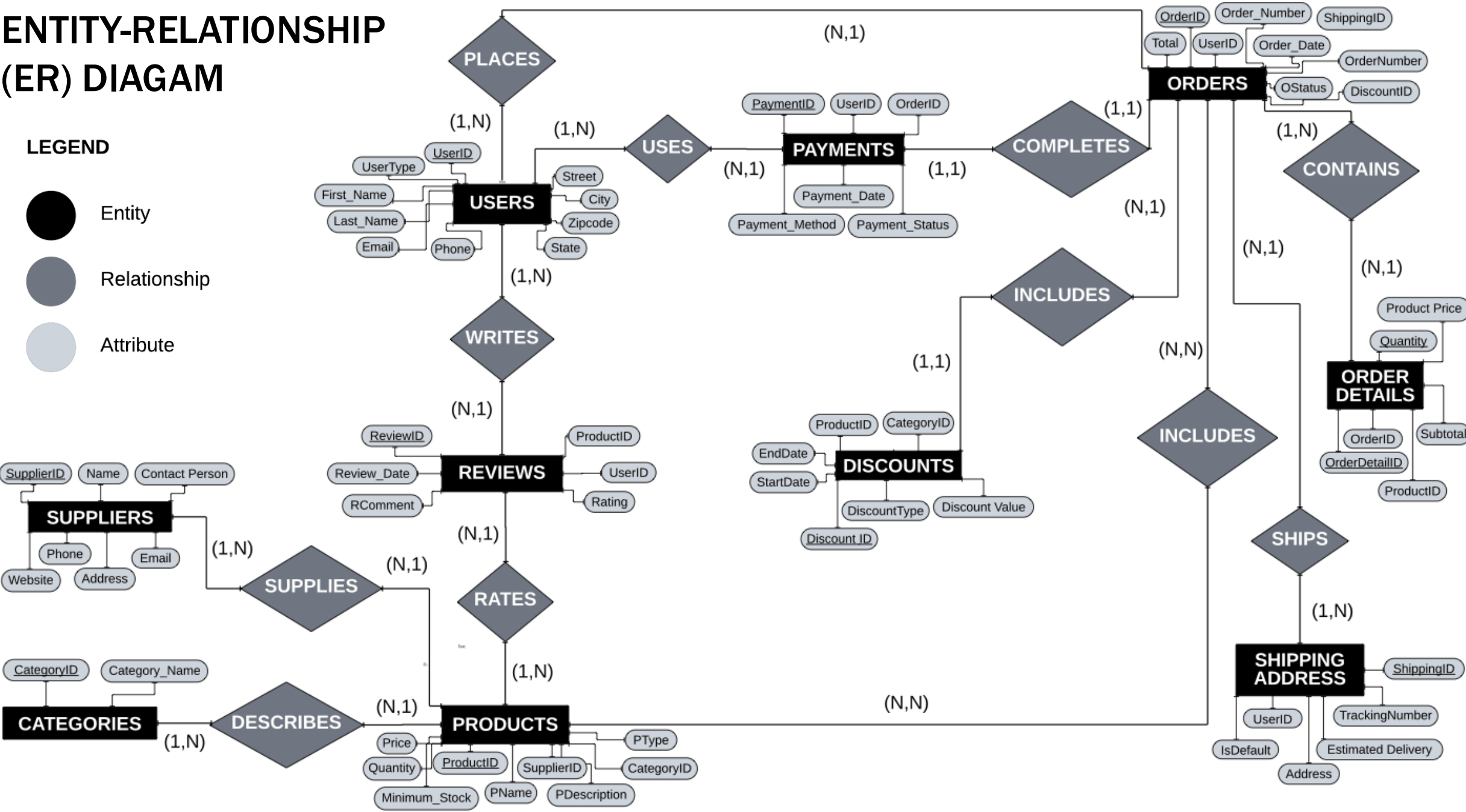
**NO** repeating groups/arrays in rows  
**NO** partial dependencies  
**NO** transitive dependencies

Categories ➡ Product ➡ Orders ➡ Users

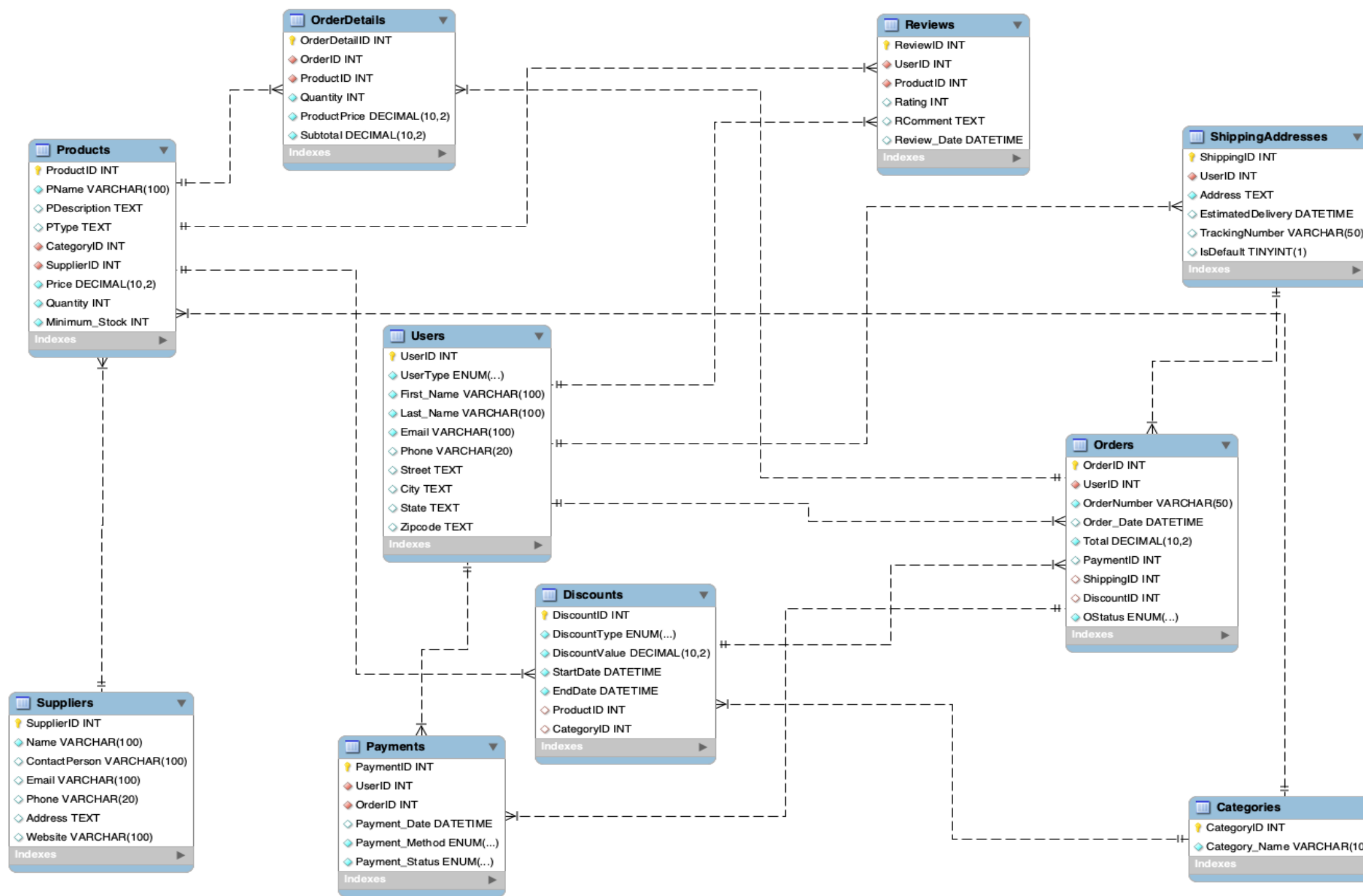
# ENTITY-RELATIONSHIP (ER) DIAGRAM

**LEGEND**

- Entity
- Relationship
- Attribute



# EER Diagram



# BHAVIN BUYS A LION

## 1. Order is created

**Enters**

User Type, Full Name, E-Mail  
Address, Phone, Password,  
Product Name, Quantity

**Generates**

User ID, Subtotal

## 3. Order is placed



**Generates**

Order ID,  
Number, Date, & Status, &  
Shipping ID, Tracking Number,  
Payment Status

## 2. Shipping & payment are updated

**Enters**

Address, Payment Method

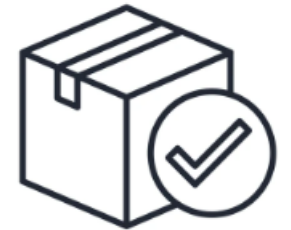
**Generates**

Total, Payment ID, Estimated  
Delivery

## 4. Order is received

**Enters**

Product ID/Name,  
Comment, Rating



**Generates**

Review ID, Review Date

# QUERIES

## Basic CRUD Queries

Adding (INSERT INTO) &  
Retrieve (SELECT) Information

## Complex Queries

Analysis (GROUP BY, ORDER BY)

- Sales Trends
- Top Sellers/Buyers

## Performance Optimization

Indexing (CREATE INDEX) for  
quick & easy retrieval of data

- Active Users
- Top Sellers & Their Description



# BASIC QUERIES - CRUD

## Create (Insert Data)

```
INSERT INTO Users (UserType, First_Name, Last_Name, Email, Phone, Street, City, State, Zipcode)  
VALUES ('Admin', 'Saeid', 'Motevali', 'smotevali1@gsu.edu', '4044132000', '33 Gilmer Street SE', 'Atlanta', 'GA', '30303');
```

## Read (Retrieve Data)

```
SELECT *  
FROM Users;
```

## Update (Modify Data)

```
UPDATE Suppliers  
SET Phone = '6789998212'  
WHERE Name = 'Fake Company';
```

## Delete (Remove Data)

```
DELETE FROM Products  
WHERE PName = 'Hamster Bedding';
```

# COMPLEX QUERIES

## FIND THE MOST TO LEAST EXPENSIVE PRODUCTS

```
# COMPLEX QUERY TO ORDER MOST EXPENSIVE ITEM TO LEAST EXPENSIVE ITEM
```

```
SELECT
```

```
    p.ProductID,  
    p.PName AS Product_Name,  
    p.CategoryID,  
    c.Category_Name,  
    p.Price,  
    p.Quantity,  
    RANK() OVER (ORDER BY p.Price DESC) AS Price_Rank
```

```
FROM
```

```
    Products p
```

```
LEFT JOIN
```

```
    Categories c ON p.CategoryID = c.CategoryID
```

```
ORDER BY
```

```
    p.Price DESC;
```

### COMPLEX SORT BY:

- EMPLOYEE RANK() FUNCTION (ADVANCED SQL FEATURE)
- DYNAMICALLY ASSIGNING RANK
- JOIN INTEGRATION – EVEN WITH DIFFERENT CATEGORICAL ITEMS
- SCALABILITY

	ProductID	Product_Name	CategoryID	Category_Name	Price	Quantity	Price_Rank
	9	Polar Bear	1	Animals	8000.00	2	1
	5	Lion	1	Animals	5000.00	10	2
	6	Bull	1	Animals	2000.00	5	3
	8	Cat	1	Animals	150.00	200	4
	1	Dog Food	5	Bedding	30.00	100	5
	7	Frog	1	Animals	25.00	300	6
	2	Cat Medicine	3	Medicine	15.00	50	7
	3	Bird Toy	4	Toys	10.00	200	8

# COMPLEX QUERIES

## FIND THE CATEGORY WITH THE HIGHEST PRODUCT AVAILABILITY

- Category 1 - with 5 Animals
- Mainly useful for admin purposes

```
# COMPLEX QUERY - GIVE THE CATEGORY WITH THE MOST PRODUCTS
```

```
WITH MostPopulatedCategory AS (  
    SELECT  
        CategoryID,  
        COUNT(*) AS Product_Count  
    FROM  
        Products  
    GROUP BY  
        CategoryID  
    ORDER BY  
        Product_Count DESC  
    LIMIT 1  
)
```

```
SELECT  
    p.CategoryID,  
    c.Category_Name,  
    p.ProductID,  
    p.PName AS Product_Name,  
    p.Price,  
    p.Quantity  
FROM  
    Products p  
JOIN  
    Categories c ON p.CategoryID = c.CategoryID  
WHERE  
    p.CategoryID = (SELECT CategoryID FROM MostPopulatedCategory)  
ORDER BY  
    p.Price DESC;
```

	CategoryID	Category_Name	ProductID	Product_Name	Price	Quantity	
	1	Animals	9	Polar Bear	8000.00	2	
	1	Animals	5	Lion	5000.00	10	
	1	Animals	6	Bull	2000.00	5	
	1	Animals	8	Cat	150.00	200	
	1	Animals	7	Frog	25.00	300	

# COMPLEX QUERIES

## FIND THE SUPPLIER WITH THE MOST PRODUCT SUPPLY

- Valuable to know what suppliers the company depends on the most
- Exclusively useful to Admin

```
# COMPLEX QUERY TO SHOW SUPPLIER WITH MOST PRODUCT DELIVERY
WITH SupplierProductCounts AS (
  SELECT
    SupplierID,
    COUNT(*) AS Total_Products
  FROM
    Products
  GROUP BY
    SupplierID
  ORDER BY
    Total_Products DESC
  LIMIT 1
)
SELECT
  p.SupplierID,
  s.Name,
  p.ProductID,
  p.PName AS Product_Name,
  p.Price,
  p.Quantity
FROM
  Products p
JOIN
  Suppliers s ON p.SupplierID = s.SupplierID
WHERE
  p.SupplierID = (SELECT SupplierID FROM SupplierProductCounts)
ORDER BY
  p.Price DESC;
```

	SupplierID	Name	ProductID	Product_Name	Price	Quantity	
	6	Exotic Petz	9	Polar Bear	8000.00	2	
	6	Exotic Petz	5	Lion	5000.00	10	
	6	Exotic Petz	6	Bull	2000.00	5	
	6	Exotic Petz	8	Cat	150.00	200	
	6	Exotic Petz	7	Frog	25.00	300	

# CREATING THE DATA

## # INSERT COMPANY SUPPLIERS

```
INSERT INTO Suppliers (Name, ContactPerson, Email, Phone, Address, Website)
VALUES
    ('Fake Company', 'Liam Maxim', 'fakeemail@gmail.com', '1234567890', '1 Fake Street, Atlanta, GA 12345', 'www.fakewebsite.com'),
    ('John Doe Supplies', 'John Doe Smith', 'johndoesupplies@gmail.com', '6795467878', '1 Happy Place Road, New York City, NY 12546', 'www.johndoesupplies.com'),
    ('Wildlife Suppliers', 'Mike Smith', 'suppliersm@example.com', '123456990', '123 Wildlife Avenue, Los Angeles, CA 30221', 'www.wildlifesuppliers.com'),
    ('Luxury Pet Supplies', 'Macy Lee', 'macylee@luxurypetsupplies.com', '0123456789', '707 Pet Lane, Atlanta, GA 54892', 'www.luxurypetsupplies.com'),
    ('Animal Superfoods', 'Jason Chin', 'jchin@animalsuperfoods.com', '4885678909', '555 Foxy Trail, Atlanta, GA 90012', 'www.animalsuperfoods.com'),
    ('Exotic Petz', 'Shown Li', 'exoticpetz@yahoo.com', '6785409897', '6960 Orlado Way Drive, Miami, FL, 34509', 'www.ebay.com/exoticpetz');
```

## # INSERTING CUSTOMERS & ADMINS

```
INSERT INTO Users (UserType, First_Name, Last_Name, Email, Phone, Street, City, State, Zipcode)
VALUES
    ('Customer', 'John', 'Doe', 'john.doe@example.com', '1234567890', '123 Main St', 'New York', 'NY', '10001'),
    ('Customer', 'Gracie', 'Rehberg', 'drehberg1@student.gsu.edu', '4789733618', '1600 Pennsylvania', 'Washington', 'DC', '20500'),
    ('Customer', 'Lilly', 'Parham', 'lparham2@student.gsu.edu', '6785487048', '6000 Universal Boulevard', 'Orlando', 'FL', '32819'),
    ('Admin', 'Saeid', 'Motevali', 'smotevali1@gsu.edu', '4044132000', '33 Gilmer Street SE', 'Atlanta', 'GA', '30303'),
    ('Customer', 'Bhavin', 'Patel', 'bpatel13@student.ksu.edu', '4784449466', '69 Rainbow Place', 'Atlanta', 'GA', '32879'),
    ('Customer', 'Selena', 'Gomez', 'selenagomez@superstar.com', '7709908989', '99 High Boulevard', 'Los Angeles', 'CA', '67678');
```

## # INSERT PRODUCT OPTIONS AVAILABLE FOR SALE

```
INSERT INTO Products (PName, PDescription, PType, CategoryID, SupplierID, Price, Quantity, Minimum_Stock)
VALUES
    ('Dog Food', 'High-protein dog food for active pets', 'Food', 5, 2, 30.00, 100, 10),
    ('Cat Medicine', 'Vitamin supplements for cats', 'Medicine', 3, 3, 15.00, 50, 5),
    ('Bird Toy', 'Colorful chewable toy for parrots', 'Toys', 4, 4, 10.00, 200, 20),
    ('Hamster Bedding', 'Soft and eco-friendly bedding for small pets', 'Bedding', 4, 5, 8.00, 300, 30),
    ('Lion', 'A large wild animal', 'Animals', 1, 6, 5000.00, 10, 1),
    ('Bull', 'A large wild animal with horns, common in Spain', 'Animals', 1, 6, 2000.00, 6, 1),
    ('Frog', 'Green jumping animal', 'Animals', 1, 6, 25.00, 300, 1),
    ('Cat', 'Friendly feline', 'Animals', 1, 6, 150.00, 200, 1),
    ('Polar Bear', 'A white, cold weather bear', 'Animals', 1, 6, 8000.00, 2, 1);
```

## # INSERT SHOPPING ITEM CATEGORIES

```
INSERT INTO Categories (Category_Name)
VALUES
    ('Animals'),
    ('Food'),
    ('Medicine'),
    ('Toys'),
    ('Bedding');
```



# DATA TABLES

[illegible][illegible]

# FUNCTIONALITY

## Triggers (CREATE TRIGGER)

- Executed after a new record is inserted into "OrderDetails" table

## Stored Procedures (CREATE PROCEDURE)

Generates...

- Total amount spent per customer
- Detailed sales report for a given period

# TRIGGERS

```
# TRIGGER WHEN ITEMS FALL BELOW RECOMMENDED MINIMUM INVENTORY LEVEL
DELIMITER $$

CREATE TRIGGER trg_update_low_stock
AFTER UPDATE ON Products
FOR EACH ROW
BEGIN
    IF NEW.Quantity < NEW.Minimum_Stock THEN
        UPDATE Products
        SET LowStockFlag = 1
        WHERE ProductID = NEW.ProductID;
    ELSE
        UPDATE Products
        SET LowStockFlag = 0
        WHERE ProductID = NEW.ProductID;
    END IF;
END$$

DELIMITER ;
```

- **Purpose:** Automatically flags products with low stock.
- **When It Runs:** After any update to the Products table.
- **What It Does:**
  - Sets LowStockFlag to 1 if Quantity is below Minimum\_Stock.
  - Sets LowStockFlag to 0 if stock is sufficient.
- **Why It's Useful:**
  - Keeps track of low-stock products in real-time.
  - Reduces the need for manual checks.
  - Uses the existing Products table — no extra tables needed.



Year

Month

Product

Total Revenue  
(\$)

% Total of  
Categories Sold

Top Sellers

% Difference of Sales  
from Previous Year

# of Products  
Sold

% Difference of Sales  
from Previous Month

Month/Year vs. Revenue

Sentiment

Average Rating

# THE INTERFACE

# THE INTERFACE

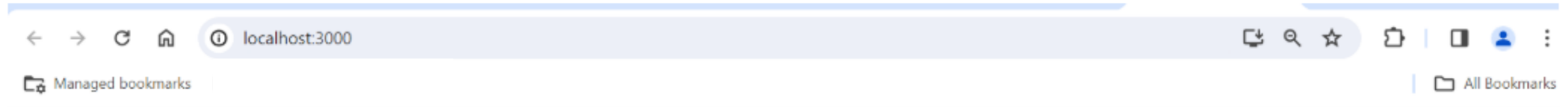
## Input Validation

- Credit Card Validation
- Address Input

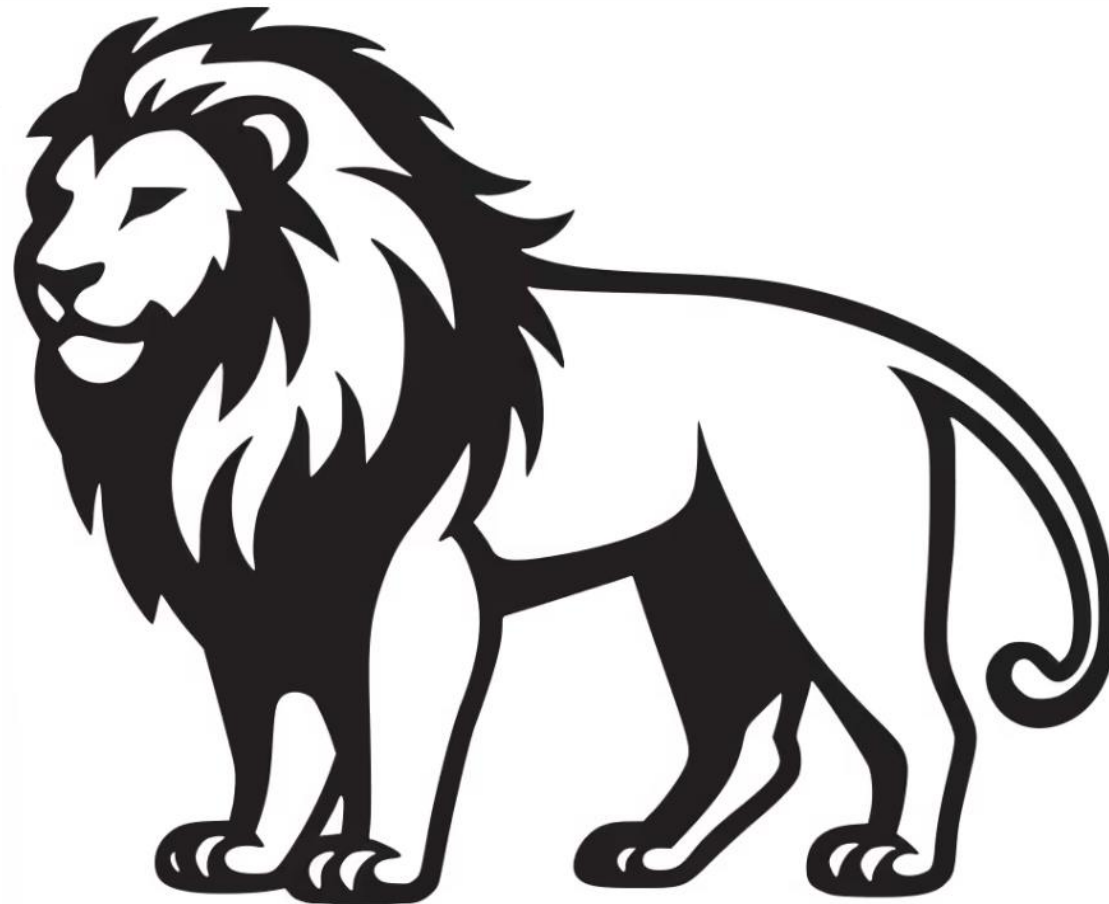
## Error Handling (User Error Messages)

- Front-end
- Back-end

# THE FRONT-END



*EXOT Animals  
E-Commerce*



With any lion  
purchase – get \$50  
off your subtotal  
until 12/31/2024!

# THE BACK-END

```
# $50 DISCOUNT ON ANY LION PURCHASES
```

```
INSERT INTO Discounts (DiscountType, DiscountValue, StartDate, EndDate, ProductID, CategoryID)
VALUES ('Flat', 50.00, NOW(), '2024-12-31', 6, 1);
```

```
# GET THE DISCOUNT ID
```

```
SELECT DiscountID
FROM Discounts
WHERE DiscountValue = 50.00 AND ProductID = 6;
```

## DISCOUNTS

- \$50 OFF ON ANY LION PURCHASE FROM NOW TO 12/31/24

```
# ADD TO ORDER DETAILS WITH DISCOUNT APPLIED
```

```
INSERT INTO OrderDetails (OrderID, ProductID, Quantity, ProductPrice, Subtotal)
```

```
VALUES (
    (SELECT OrderID FROM Orders WHERE OrderNumber = 'ORD10001'),
    6,
    1,
    5000.00,
    5000.00 - (SELECT DiscountValue
                FROM Discounts
                WHERE ProductID = 6 AND CategoryID = 1)
);
```

```
# UPDATE TOTAL IN ORDERS TO REFLECT DISCOUNT
```

```
UPDATE Orders
```

```
SET Total = Total - (SELECT DiscountValue
                     FROM Discounts
                     WHERE ProductID = 6 AND CategoryID = 1)
WHERE OrderNumber = 'ORD10001';
```

# THE FRONT-END

## *Exot Animals E-Commerce*

StoreFront

Your Cart



[View Details](#)



[View Details](#)



[View Details](#)



[View Details](#)



[View Details](#)



[View Details](#)



[View Details](#)



[View Details](#)



[View Details](#)



[View Details](#)

# THE BACK-END

## ORDERS

- BHAVIN IS OPENING A NEW ZOO, HE STILL NEEDS A LION AND SEES THEY ARE ON DISCOUNT AT EXOT ANIMALS E-COMMERCE, SO HE DECIDES TO ORDER ONE.

```
# USER 5 (BHAVIN PATEL) WANTS TO BUY A LION – HE STARTS THE ORDER
INSERT INTO Orders (UserID, OrderNumber, Total, PaymentID, ShippingID, DiscountID, OStatus)
VALUES (5, 'ORD10001', 5000.00, NULL, NULL, NULL, 'Pending');

# GIVES US THE ORDER ID
SELECT OrderID
FROM Orders
WHERE OrderNumber = 'ORD10001';
```


# THE FRONT-END

## *Exot Animals E-Commerce*

### Payment Information

- ☒ Credit Card    
- ☐ PayPal [What's this?](#)
- ☐ venmo [What's this?](#)
- ☐ amazon pay [What's this?](#)

#### PAY BY CREDIT/DEBIT CARD

 Secure payment with 128-bit SSL encryption technology



Card Number :

Expiration Date:  Month  Year

CVV2 Number:  [What's this?](#)

### Billing Information

☐ Same as John Newman

Name:  First Name  Last Name

Company:

Address1:

Address2:

Country:  United States

Zip/Postal Code:

City/Town:

State/Province:  Please Choose...

Please provide at least one phone number

Cell Phone:

Home Phone:

Work Phone:

P.O. #/Other Ref:

Email Address:

We will send this email address an Order Confirmation.



# THE BACK-END

```
# BHAVIN PAYS FOR THE LION - ORDER # ORD10001
INSERT INTO Payments (UserID, OrderID, Payment_Date, Payment_Method, Payment_Status)
VALUES (5, (SELECT OrderID FROM Orders WHERE OrderNumber = 'ORD10001'), NOW(), 'Credit Card', 'Completed');

# GETS THE PAYMENT ID FOR THE TRANSACTION
SELECT PaymentID
FROM Payments
WHERE OrderID = (SELECT OrderID FROM Orders WHERE OrderNumber = 'ORD10001');

# INSERTS BHAVIN'S SHIPPING LOCATION
INSERT INTO ShippingAddresses (UserID, Address, EstimatedDelivery, TrackingNumber, IsDefault)
VALUES (5, '789 New Street, Los Angeles, CA, 90001', '2025-01-15', 'SHIP123456789', TRUE);

# GET THE SHIPPING ID
SELECT ShippingID
FROM ShippingAddresses
WHERE UserID = 5
AND Address = '789 New Street, Los Angeles, CA, 90001';
```

```
# SET ORDER STATUS TO SHIPPED AFTER PAYMENT IS COMPLETED
UPDATE Orders
SET OStatus = 'Shipped'
WHERE OrderNumber = 'ORD10001'
AND PaymentID = (SELECT PaymentID FROM Payments WHERE OrderID = Orders.OrderID AND Payment_Status = 'Completed');
```

## PAYMENT & SHIPPING

- BHAVIN MUST PAY FOR THE LION. HIS CREDIT CARD IS VETTED ON THE BACKEND, ENSURING HE HAS THE NECESSARY FUNDS, THEN HE IS ALLOWED TO ENTER HIS SHIPPING INFO.

# BHAVIN'S ORDER INFO.

	OrderID	UserID	OrderNumber	Order_Date	Total	PaymentID	Shi...	DiscountID	OStatus	
	1	5	ORD10001	2024-12-04 12:40:37	4950.00	1	1	1	Shipped	
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

	Pay...	UserID	OrderID	Payment_Date	Payment_Method	Payment_Status	
	1	5	1	2024-12-04 12:40:47	Credit Card	Completed	
	NULL	NULL	NULL	NULL	NULL	NULL	

	Shi...	UserID	Address	EstimatedDelivery	TrackingNumb...	IsDefault	
	1	5	789 New Street, Los Angeles, CA, 90001	2025-01-15 00:00:00	SHIP123456789	1	
	NULL	NULL	NULL	NULL	NULL	NULL	

# APPENDICES

# **APPENDIX I**

## **Data Dictionary**

# ENTITY: USERS

Attribute	Data Type	Description
UserID	Integer	Unique identifier for the user.
Name	String	Full name of the user.
Email	String	Email address of the user.
Password	String	Encrypted password for user authentication.
Address	String	Physical address of the user.
Phone	String	Contact phone number of the user.
UserType	Enum (Admin/Customer)	Specifies whether the user is an admin or a customer.

# ENTITY: PRODUCTS

Attribute	Data Type	Description
ProductID	Integer	Unique identifier for the product.
Name	String	Name of the product.
CategoryID	Integer (FK)	Identifier linking to the product category.
Price	Float	Price of the product.
StockQuantity	Integer	Number of units available in stock.
Description	String	Detailed description of the product.
SupplierID	Integer (FK)	Identifier linking to the supplier of the product.

# ENTITY: CATEGORIES

Attribute	Data Type	Description
CategoryID	Integer	Unique identifier for the category
CategoryName	String	Name of the category

# ENTITY: ORDERS

Attribute	Data Type	Description
OrderID	Integer	Unique identifier for the order.
UserID	Integer (FK)	Identifier linking to the user who placed the order.
OrderDate	Date	Date the order was placed.
TotalAmount	Float	Total cost of the order.
PaymentID	Integer (FK)	Identifier linking to the payment details.
Status	Enum (Pending/Completed/Cancelled)	Current status of the order.
ShippingID	Integer (FK)	Identifier linking to shipping details.
DiscountID	Integer (FK)	Identifier linking to applied discounts.



# ENTITY: ORDER DETAILS

Attribute	Data Type	Description
OrderDetailID	Integer	Unique identifier for the order detail.
OrderID	Integer (FK)	Identifier linking to the order.
OrderNumber	String	Reference number for the order.
ProductID	Integer (FK)	Identifier linking to the product.
Quantity	Integer	Quantity of the product in the order.
ProductPrice	Float	Price of the product at the time of the order.

# ENTITY: REVIEWS

Attribute	Data Type	Description
ReviewID	Integer	Unique identifier for the review.
ProductID	Integer (FK)	Identifier linking to the reviewed product.
UserID	Integer (FK)	Identifier linking to the reviewer.
Rating	Integer (1-5)	Rating given by the user.
Comment	String	Review comments.
ReviewDate	Date	Date of the review.

# ENTITY: PAYMENTS

Attribute	Data Type	Description
PaymentID	Integer	Unique identifier for the payment.
UserID	Integer (FK)	Identifier linking to the user.
OrderID	Integer (FK)	Identifier linking to the order.
PaymentMethod	Enum (Credit Card/PayPal/Bank Transfer)	Method of payment.
PaymentStatus	Enum (Successful/Failed/Pending)	Status of the payment.
PaymentDate	Date	Date

# ENTITY: SHIPPING ADDRESSES

Attribute Name	Data Type	Description
ShippingID	Integer	Unique identifier for each shipping address.
UserID	Integer	Foreign key referencing Users table.
Address	String	Shipping address.
EstimatedDelivery	Date	Estimated delivery date.
TrackingNumber	String	Tracking number for the shipment.

# ENTITY: DISCOUNTS

Attribute Name	Data Type	Description
DiscountID	Integer	Unique identifier for each discount.
DiscountType	String	Type of discount (e.g., Percentage).
DiscountValue	Float	Value of the discount.
StartDate	Date	Start date of the discount.
EndDate	Date	End date of the discount.
ProductID	Integer	Foreign key referencing Products table.
CategoryID	Integer	Foreign key referencing Categories table.

# ENTITY: SUPPLIERS

Attribute Name	Data Type	Description
SupplierID	Integer	Unique identifier for each supplier.
Name	String	Name of the supplier.
ContactPerson	String	Primary contact person at the supplier.
Email	String	Email address for the supplier.
Phone	String	Contact phone number for the supplier.
Address	String	Address of the supplier.
Website	String	Supplier's website URL.

# APPENDIX II

EER Diagram

## Reviews

- ReviewID INT
- UserID INT
- ProductID INT
- Rating INT
- RComment TEXT
- Review\_Date DATETIME

Indexes

## OrderDetails

- OrderDetailID INT
- OrderID INT
- ProductID INT
- Quantity INT
- ProductPrice DECIMAL(10,2)
- Subtotal DECIMAL(10,2)

Indexes

## Products

- ProductID INT
- PName VARCHAR(100)
- PDescription TEXT
- PType TEXT
- CategoryID INT
- SupplierID INT
- Price DECIMAL(10,2)
- Quantity INT
- Minimum\_Stock INT

Indexes





## Discounts



- 🔑 DiscountID INT
- 💠 DiscountType ENUM(...)
- 💠 DiscountValue DECIMAL(10,2)
- 💠 StartDate DATETIME
- 💠 EndDate DATETIME
- 🔻 ProductID INT
- 🔻 CategoryID INT

Indexes



## Suppliers



- 🔑 SupplierID INT
- 💠 Name VARCHAR(100)
- 💠 ContactPerson VARCHAR(100)
- 💠 Email VARCHAR(100)
- 💠 Phone VARCHAR(20)
- 💠 Address TEXT
- 💠 Website VARCHAR(100)

Indexes



## Users



- 🔑 UserID INT
- 💠 UserType ENUM(...)
- 💠 First\_Name VARCHAR(100)
- 💠 Last\_Name VARCHAR(100)
- 💠 Email VARCHAR(100)
- 💠 Phone VARCHAR(20)
- 💠 Street TEXT
- 💠 City TEXT
- 💠 State TEXT
- 💠 Zipcode TEXT

Indexes





## ShippingAddresses



- ShippingID INT
- UserID INT
- Address TEXT
- EstimatedDelivery DATETIME
- TrackingNumber VARCHAR(50)
- IsDefault TINYINT(1)

### Indexes



## Categories



- CategoryID INT
- Category\_Name VARCHAR(100)

### Indexes



## Orders



- OrderID INT
- UserID INT
- OrderNumber VARCHAR(50)
- Order\_Date DATETIME
- Total DECIMAL(10,2)
- PaymentID INT
- ShippingID INT
- DiscountID INT
- OStatus ENUM(...)

### Indexes



## Payments



- PaymentID INT
- UserID INT
- OrderID INT
- Payment\_Date DATETIME
- Payment\_Method ENUM(...)
- Payment\_Status ENUM(...)

### Indexes

