

**NEXON ECOMMARCE WEBSITE**

**SUBMITTED BY**

MUHAMMAD SARIM F2021105128

QURRAT UL AIN F2021105129

DAUD MUSTAFA F2021105159

MUHAMMAD AWAIS F2021105177

**SUBMITTED TO**

Ms. Taila Jabeen

**Subject**

DATABASE SYSTEMS

**Section**

Y-2

1. **Scope**
   1. **Overview:** Develop an Ecommerce website specifically designed for the sale and distribution of documentation-related products. The website will provide a platform for users to browse, search, purchase, and download various types of documentation, such as user manuals. The technical guides, instructional materials, and more.
   2. **Features:**

a. **User Registration and Authentication**:

Users can create accounts with their email addresses.

Users can log in using their credentials.

Password reset functionality will be implemented.

**b. Product Browsing:**

Users can browse products by categories.

Products will display their name, price, description, and production year.

Users can view detailed product information.

c**. Shopping Cart:**

Users can add products to their shopping cart.

Users can specify the quantity of each product.

Total cost calculation will be handled accurately.

**d. Checkout Process:**

Users can proceed to the checkout process.

Users can provide their delivery information (address, contact number).

Users can select a payment method.

**e. Order Placement:**

Orders will be created after successful checkout.

Orders will include the customer's information, purchased products, payment details, and delivery information.

**f. Order History:**

Users can view their order history.

Users can see the status of their orders.

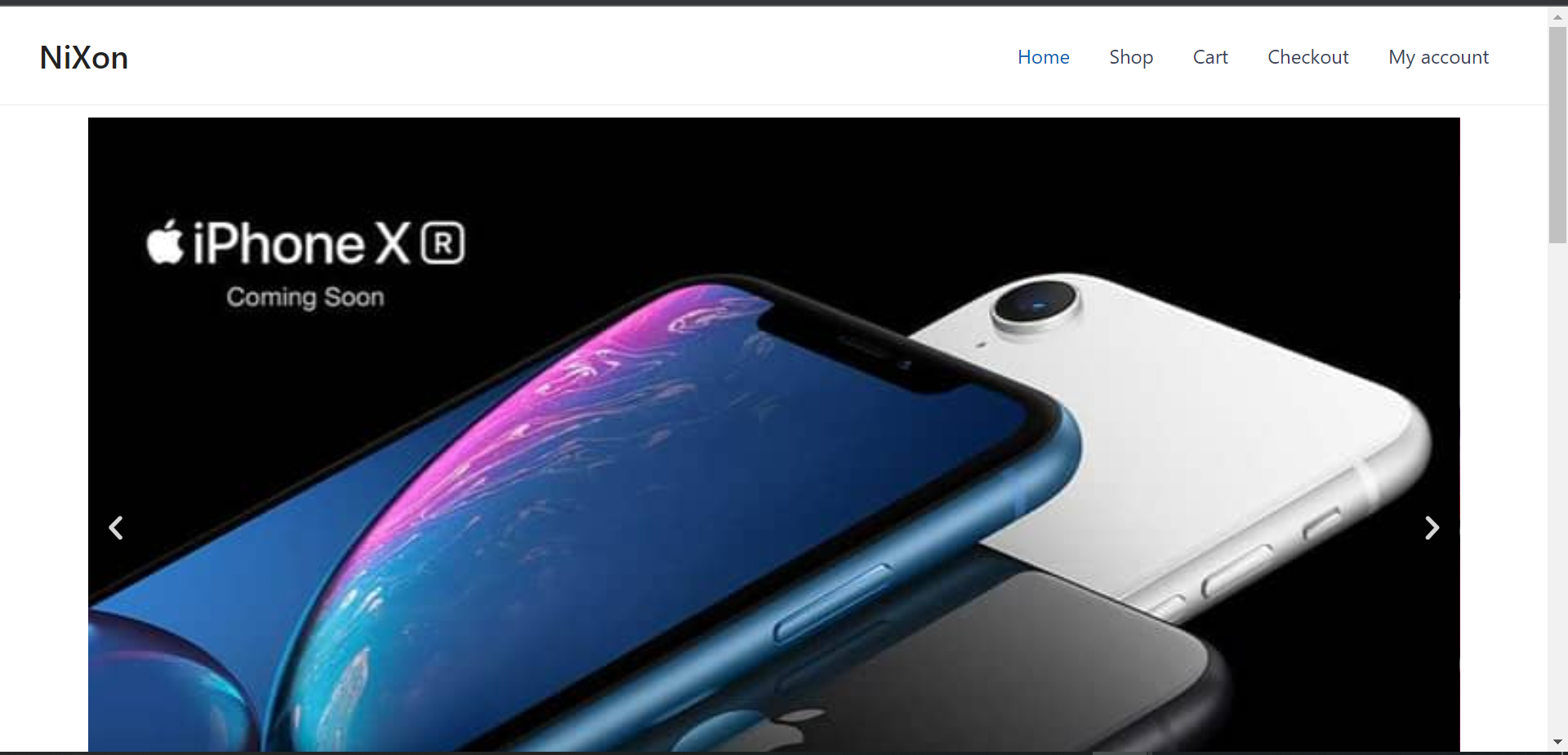
**g. Customer Support:**

Users can contact customer support for assistance.

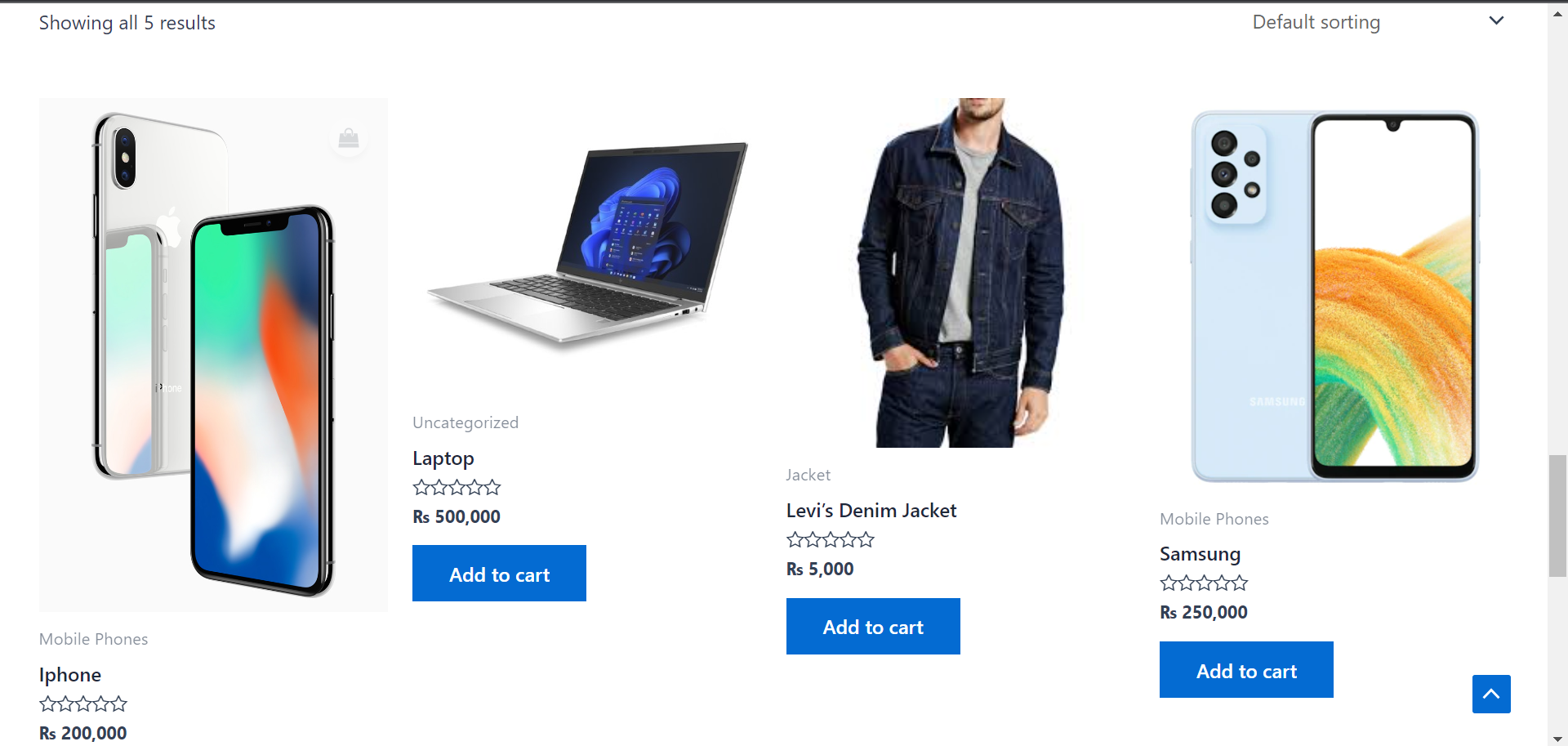
Users can provide feedback or raise concerns.

**Snapshots (user Interface)**

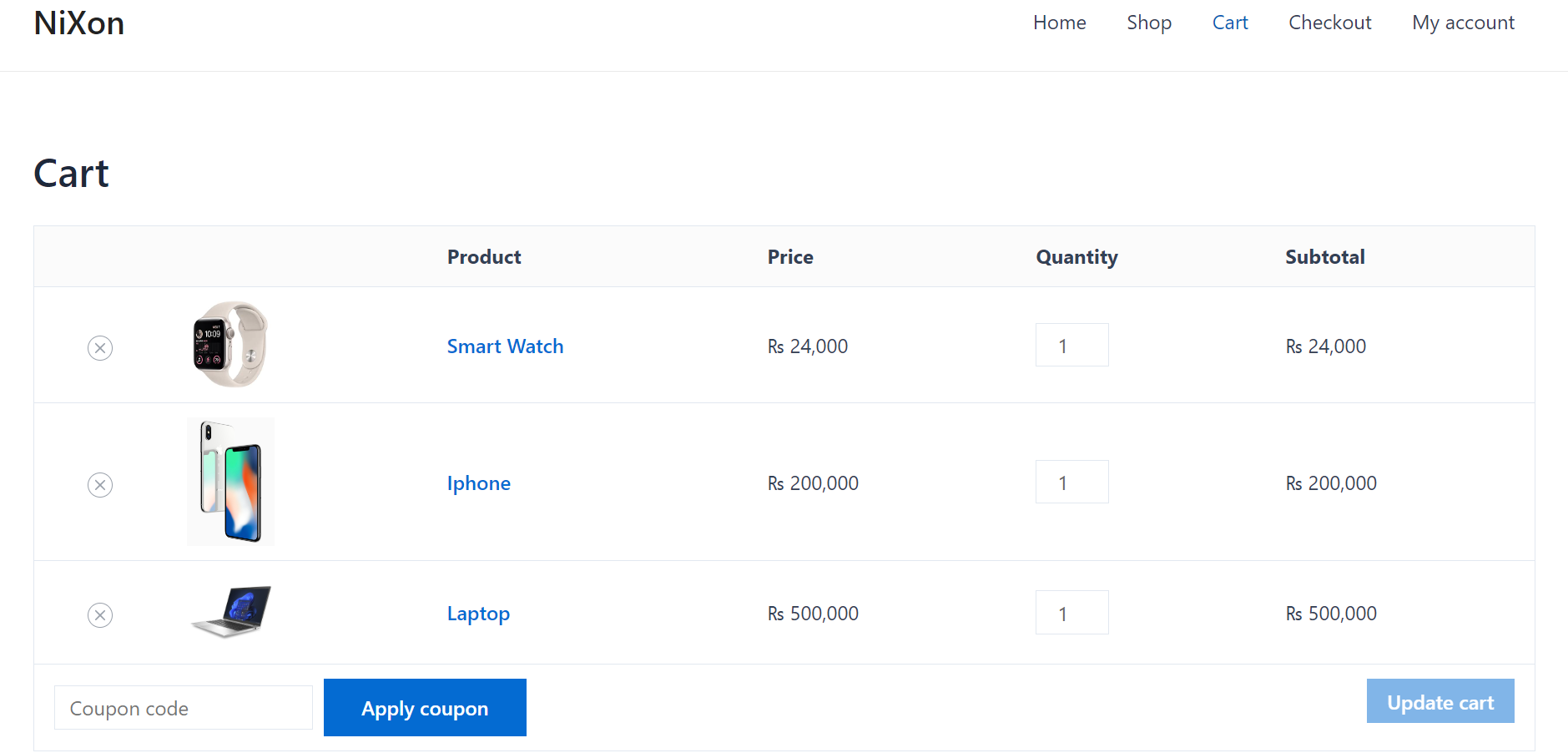
Home Webpage



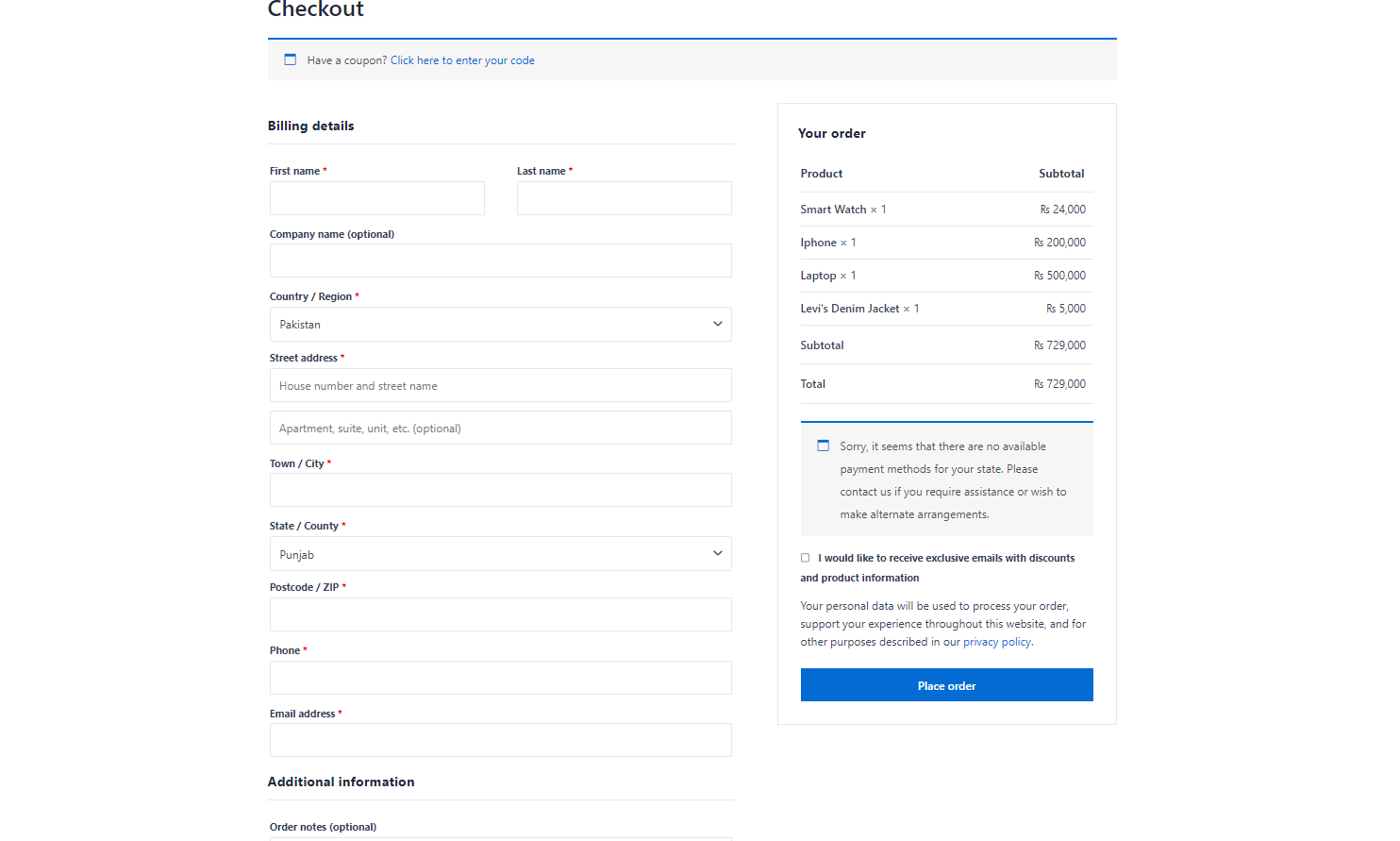
Shop page



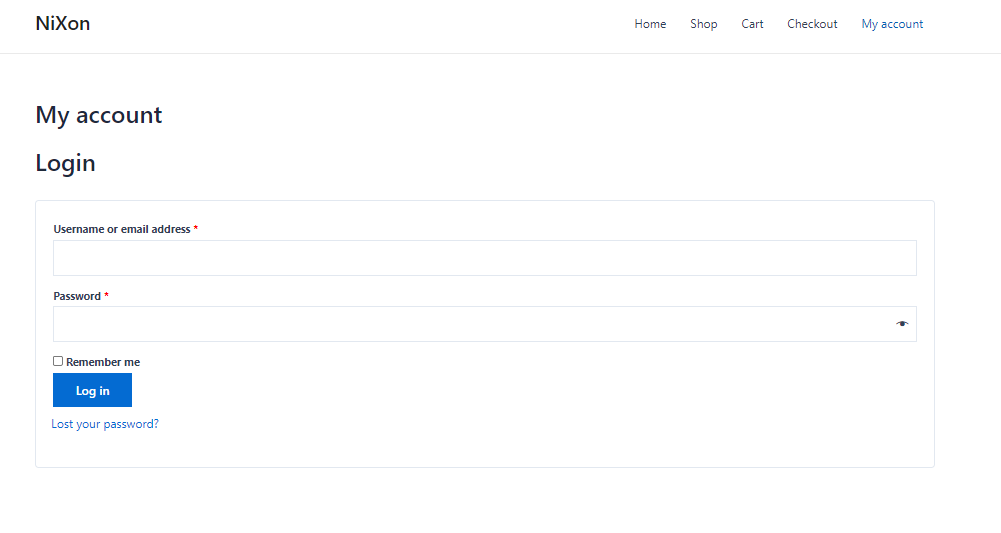
**Cart Page**

****

Check out



My Account Page



**Entities:**

* Entities represent real-world objects or concepts in a database.
* They can be tangible things like a person, a car, or a product, or intangible concepts like an order or a reservation.
* Each entity typically has a unique identifier called a primary key, which distinguishes it from other entities of the same type.
* Examples of entities: Customer, Employee, Product, Order.

**Attributes:**

* Attributes are characteristics or properties of entities.
* They describe the qualities or features of an entity.
* Each entity has one or more attributes that define its properties.
* Attributes can be simple or composite (composed of sub-attributes).
* They can be single-valued (containing one value) or multi-valued (containing multiple values).
* Examples of attributes: Name, Age, Address, Email.

**ENTITIES AND ATTRIBUTES IN E-COMMERCE WEB**

**Category Entity:**

Category\_ID (Primary Key)

Category\_Name

**Customer Entity:**

Customer\_ID (Primary Key)

Customer\_Email

Customer\_FirstName

Customer\_LastName

Address\_House\_Number

Address\_Postal\_Code

Address\_Province

Address\_City

**Customer\_ContactNo Entity:**

Contact\_Number

Customer\_ID (Foreign Key referencing Customer(Customer\_ID))

**Payment Entity:**

Payment\_ID (Primary Key)

Customer\_ID (Foreign Key referencing Customer(Customer\_ID))

Payment\_Date

Payment\_Amount

Payment\_Method

**Product Entity:**

Product\_ID (Primary Key)

Product\_Name

Product\_Price

Product\_Description

Production\_Year

Category\_ID (Foreign Key referencing Category(Category\_ID))

Customer\_ID (Foreign Key referencing Customer(Customer\_ID))

**Cart Entity:**

Cart\_ID (Primary Key)

Total\_Cost

Product\_ID (Foreign Key referencing Product(Product\_ID))

Total\_Product\_Quantity

**Courier Entity:**

Courier\_ID (Primary Key)

Courier\_Name

Expected\_Delivery\_Date

Current\_Location

Cart\_ID (Foreign Key referencing Cart(Cart\_ID))

Customer\_ID (Foreign Key referencing Customer(Customer\_ID))

**Courier\_ContactNo Entity:**

Contact\_No

Courier\_ID (Foreign Key referencing Courier(Courier\_ID))

DeliveryTo Entity:

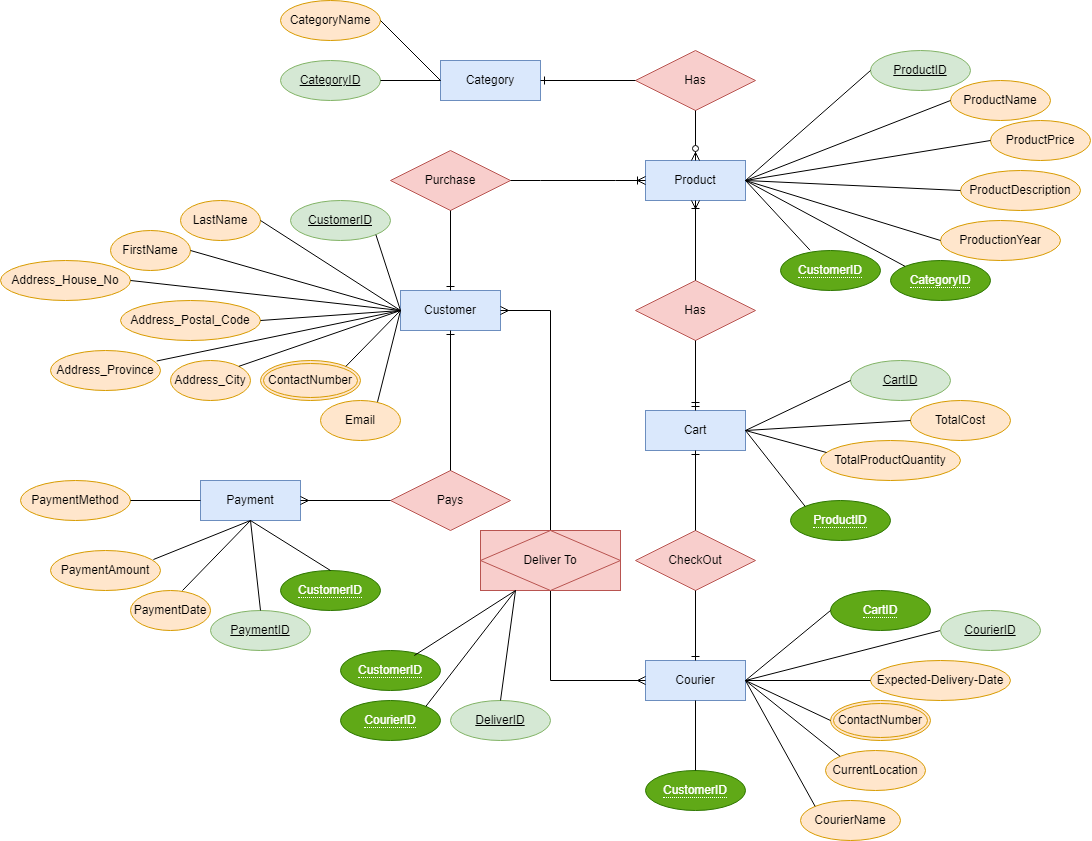
Delivery\_ID (Primary Key)

Delivery\_Type

Customer\_ID (Foreign Key referencing Customer(Customer\_ID))

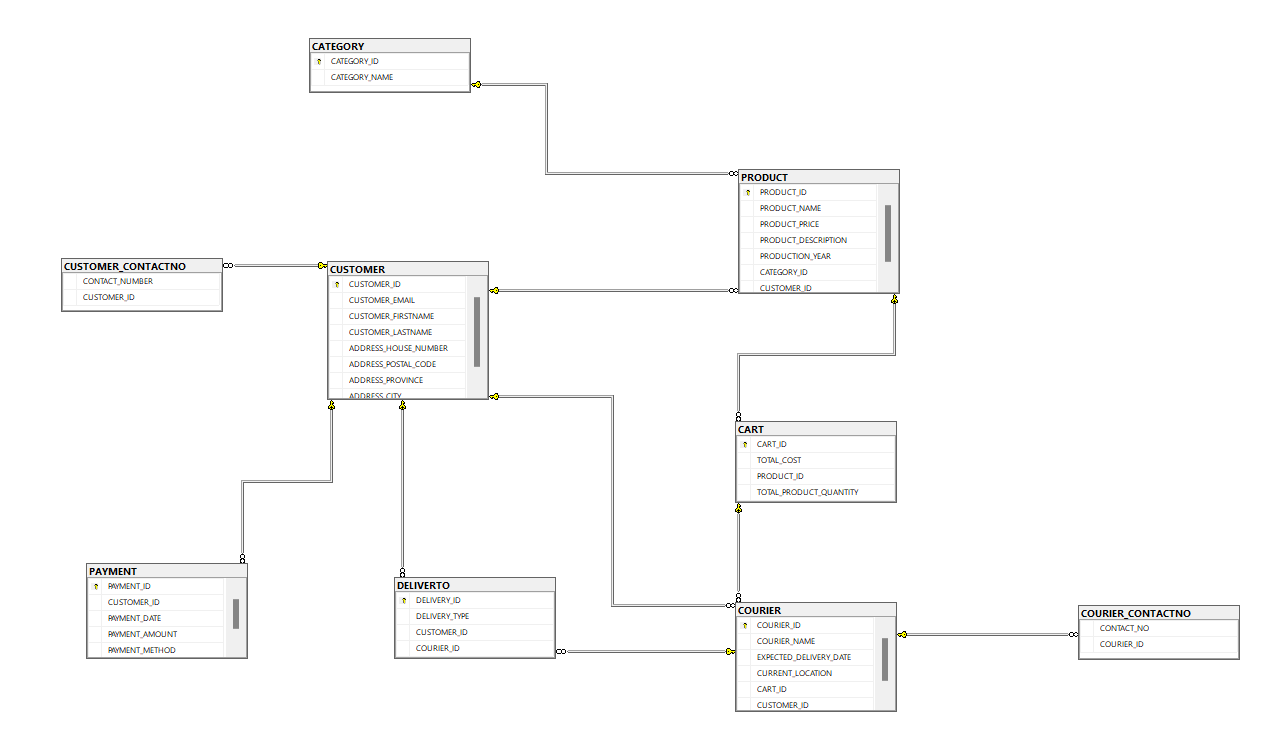
Courier\_ID (Foreign Key referencing Courier(Courier\_ID))

**E-R DIAGRAM**



* **ER diagram stands for Entity-Relationship diagram.**
* It is a visual representation of the entities (objects or concepts) and their relationships in a database.
* The diagram consists of entities, attributes, and relationships.
* Entities represent real-world objects or concepts, such as a person, a product, or a customer.
* Attributes are properties or characteristics of entities, such as name, age, or price.
* Relationships define associations between entities, indicating how they are related to each other.
* Relationships can be one-to-one, one-to-many, or many-to-many.
* Cardinality and participation constraints can be represented in the diagram.
* Cardinality specifies the number of instances of one entity that can be associated with another entity.
* Participation constraints determine whether the participation of an entity in a relationship is mandatory (total) or optional (partial).
* The ER diagram serves as a blueprint for designing the database schema.
* It helps in identifying the primary keys, foreign keys, and the overall structure of the database.

**Relational Diagram**



* **Relational diagram** represents the logical structure of a relational database.
* Tables are depicted as rectangles or boxes in the diagram.
* Each table represents an entity or concept in the database.
* The name of the table is usually displayed at the top of the rectangle.
* Attributes of a table are displayed inside the rectangle, usually as a list of columns.
* Primary keys are indicated in the diagram to uniquely identify each record in a table.
* Relationships between tables are represented by lines connecting them.
* The lines indicate the nature of the relationship, such as one-to-one, one-to-many, or many-to-many.
* Foreign keys, which establish the connections between tables, are depicted in the diagram.
* Cardinality and participation constraints can be specified in the diagram to indicate the nature of the relationships.
* The relational diagram provides an overview of the database structure and aids in understanding the relationships between tables and the attributes within them.
* It is used during the database design phase to plan and organize the database schema.

**SQL**

CREATE DATABASE ECOMMERCE

CREATE TABLE CATEGORY(

CATEGORY\_ID INT PRIMARY KEY NOT NULL,

CATEGORY\_NAME VARCHAR(50) NOT NULL

)

CREATE TABLE CUSTOMER(

CUSTOMER\_ID INT PRIMARY KEY NOT NULL,

CUSTOMER\_EMAIL VARCHAR(50) NOT NULL,

CUSTOMER\_FIRSTNAME VARCHAR(50) NOT NULL,

CUSTOMER\_LASTNAME VARCHAR(50),

ADDRESS\_HOUSE\_NUMBER VARCHAR(100) NOT NULL,

ADDRESS\_POSTAL\_CODE VARCHAR(50) NOT NULL,

ADDRESS\_PROVINCE VARCHAR(30) NOT NULL,

ADDRESS\_CITY VARCHAR(30) NOT NULL,

)

CREATE TABLE CUSTOMER\_CONTACTNO(

CONTACT\_NUMBER VARCHAR(20),

CUSTOMER\_ID INT FOREIGN KEY(CUSTOMER\_ID) REFERENCES CUSTOMER(CUSTOMER\_ID)

)

CREATE TABLE PAYMENT(

PAYMENT\_ID INT PRIMARY KEY NOT NULL,

CUSTOMER\_ID INT FOREIGN KEY(CUSTOMER\_ID) REFERENCES CUSTOMER(CUSTOMER\_ID),

PAYMENT\_DATE DATE NOT NULL,

PAYMENT\_AMOUNT MONEY NOT NULL,

PAYMENT\_METHOD VARCHAR(20) NOT NULL

)

CREATE TABLE PRODUCT(

PRODUCT\_ID INT PRIMARY KEY NOT NULL,

PRODUCT\_NAME VARCHAR(50) NOT NULL,

PRODUCT\_PRICE MONEY CHECK(PRODUCT\_PRICE>0) NOT NULL,

PRODUCT\_DESCRIPTION VARCHAR(100),

PRODUCTION\_YEAR DATE,

CATEGORY\_ID INT FOREIGN KEY(CATEGORY\_ID) REFERENCES CATEGORY(CATEGORY\_ID),

CUSTOMER\_ID INT FOREIGN KEY(CUSTOMER\_ID) REFERENCES CUSTOMER(CUSTOMER\_ID)

)

CREATE TABLE CART(

CART\_ID INT PRIMARY KEY NOT NULL,

TOTAL\_COST MONEY NOT NULL,

PRODUCT\_ID INT FOREIGN KEY(PRODUCT\_ID) REFERENCES PRODUCT(PRODUCT\_ID),

TOTAL\_PRODUCT\_QUANTITY INT CHECK(TOTAL\_PRODUCT\_QUANTITY>0) NOT NULL

)

CREATE TABLE COURIER(

COURIER\_ID INT PRIMARY KEY NOT NULL,

COURIER\_NAME VARCHAR(50) NOT NULL,

EXPECTED\_DELIVERY\_DATE DATE NOT NULL,

CURRENT\_LOCATION VARCHAR(50) NOT NULL,

CART\_ID INT FOREIGN KEY(CART\_ID) REFERENCES CART(CART\_ID),

CUSTOMER\_ID INT FOREIGN KEY(CUSTOMER\_ID) REFERENCES CUSTOMER(CUSTOMER\_ID)

)

CREATE TABLE COURIER\_CONTACTNO(

CONTACT\_NO VARCHAR(20),

COURIER\_ID INT FOREIGN KEY(COURIER\_ID) REFERENCES COURIER(COURIER\_ID)

)

CREATE TABLE DELIVERTO(

DELIVERY\_ID INT PRIMARY KEY NOT NULL,

DELIVERY\_TYPE VARCHAR(50) NOT NULL,

CUSTOMER\_ID INT FOREIGN KEY(CUSTOMER\_ID) REFERENCES CUSTOMER(CUSTOMER\_ID),

COURIER\_ID INT FOREIGN KEY(COURIER\_ID) REFERENCES COURIER(COURIER\_ID)

)

**INSERT DATA**

INSERT INTO CATEGORY(CATEGORY\_ID,CATEGORY\_NAME)

VALUES

(1, 'Electronics'),

(2, 'Clothing'),

(3, 'Home & Kitchen'),

(4, 'Books'),

(5, 'Sports & Outdoors'),

(6, 'Beauty'),

(7, 'Toys & Games'),

(8, 'Automotive'),

(9, 'Health & Personal Care'),

(10, 'Jewelry')

INSERT INTO CUSTOMER (CUSTOMER\_ID, CUSTOMER\_EMAIL, CUSTOMER\_FIRSTNAME, CUSTOMER\_LASTNAME, ADDRESS\_HOUSE\_NUMBER, ADDRESS\_POSTAL\_CODE, ADDRESS\_PROVINCE, ADDRESS\_CITY)

VALUES

(1, 'john@example.com', 'John', 'Doe', '123 Main S', '12345', 'California', 'Los Angeles'),

(2, 'jane@example.com', 'Jane', 'Smith', '456 Oak', '67890', 'New York', 'New York City'),

(3, 'mike@example.com', 'Mike', 'Johnson', '789 Elm', '54321', 'Texas', 'Houston'),

(4, 'amy@example.com', 'Amy', 'Williams', '987 Pine', '13579', 'Florida', 'Miami'),

(5, 'david@example.com', 'David', 'Brown', '321 Cedar', '86420', 'Illinois', 'Chicago'),

(6, 'emily@example.com', 'Emily', 'Taylor', '555 Maple', '24680', 'Washington', 'Seattle'),

(7, 'alex@example.com', 'Alex', 'Davis', '222 Walnut', '97531', 'Massachusetts', 'Boston'),

(8, 'lisa@example.com', 'Lisa', 'Martinez', '444 Oak', '68390', 'Colorado', 'Denver'),

(9, 'sam@example.com', 'Sam', 'Anderson', '777 Elm', '19567', 'Arizona', 'Phoenix'),

(10, 'sarah@example.com', 'Sarah', 'Garcia', '888 Pine', '43785', 'Georgia', 'Atlanta'),

(11, 'mark@example.com', 'Mark', 'Wilson', '111 Oak', '24680', 'California', 'San Francisco'),

(12, 'laura@example.com', 'Laura', 'Thompson', '222 Elm', '13579', 'New York', 'Albany'),

(13, 'chris@example.com', 'Chris', 'Walker', '333 Maple', '86420', 'Texas', 'Dallas'),

(14, 'jennifer@example.com', 'Jennifer', 'Harris', '444 Pine', '97531', 'Florida', 'Orlando'),

(15, 'kevin@example.com', 'Kevin', 'Lewis', '555 Cedar', '68390', 'Illinois', 'Springfield'),

(16, 'sophia@example.com', 'Sophia', 'Clark', '666 Walnut', '19567', 'Washington', 'Spokane'),

(17, 'ryan@example.com', 'Ryan', 'Young', '777 Oak', '43785', 'Massachusetts', 'Worcester'),

(18, 'natalie@example.com', 'Natalie', 'Hall', '888 Elm', '36987', 'Colorado', 'Colorado Springs'),

(19, 'peter@example.com', 'Peter', 'Allen', '999 Pine', '58291', 'Arizona', 'Tucson'),

(20, 'olivia@example.com', 'Olivia', 'Turner', '101 Maple', '71936', 'Georgia', 'Savannah'),

(21, 'robert@example.com', 'Robert', 'Parker', '111 Walnu', '49572', 'California', 'San Diego'),

(22, 'amanda@example.com', 'Amanda', 'Phillips', '222 Elm', '72648', 'New York', 'Buffalo'),

(23, 'michael@example.com', 'Michael', 'Roberts', '333 Cedar', '28573', 'Texas', 'Austin'),

(24, 'jessica@example.com', 'Jessica', 'Scott', '444 Pine', '61749', 'Florida', 'Tampa'),

(25, 'andrew@example.com', 'Andrew', 'Adams', '555 Oak', '39482', 'Illinois', 'Peoria'),

(26, 'grace@example.com', 'Grace', 'Walker', '666 Elm', '95836', 'Washington', 'Bellevue'),

(27, 'daniel@example.com', 'Daniel', 'Morris', '777 Maple', '26194', 'Massachusetts', 'Cambridge'),

(28, 'samantha@example.com', 'Samantha', 'Baker', '888 Walnut', '67934', 'Colorado', 'Boulder'),

(29, 'jonathan@example.com', 'Jonathan', 'Mitchell', '999 Pine', '17485', 'Arizona', 'Mesa'),

(30, 'hannah@example.com', 'Hannah', 'Hayes', '101 Cedar', '62739', 'Georgia', 'Augusta'),

(31, 'matthew@example.com', 'Matthew', 'Ward', '111 Oak', '53642', 'California', 'San Jose'),

(32, 'victoria@example.com', 'Victoria', 'Rogers', '222 Elm', '94638', 'New York', 'Rochester'),

(33, 'ethan@example.com', 'Ethan', 'Sullivan', '333 Maple', '28194', 'Texas', 'Houston'),

(34, 'lily@example.com', 'Lily', 'Cooper', '444 Pine', '76391', 'Florida', 'Miami'),

(35, 'benjamin@example.com', 'Benjamin', 'Turner', '555 Cedar', '16285', 'Illinois', 'Chicago'),

(36, 'audrey@example.com', 'Audrey', 'Perry', '666 Walnut', '87462', 'Washington', 'Seattle'),

(37, 'nathan@example.com', 'Nathan', 'Kelly', '777 Oak', '53624', 'Massachusetts', 'Boston'),

(38, 'claire@example.com', 'Claire', 'Barnes', '888 Elm', '27394', 'Colorado', 'Denver'),

(39, 'jacob@example.com', 'Jacob', 'Wood', '999 Pine', '74938', 'Arizona', 'Phoenix'),

(40, 'zoey@example.com', 'Zoey', 'Turner', '101 Maple', '47593', 'Georgia', 'Atlanta'),

(41, 'sebastian@example.com', 'Sebastian', 'Morgan', '111 Walnut', '94627', 'California', 'San Francisco'),

(42, 'sean@example.com', 'Sean', 'Turner', '444 Pine', '97462', 'Florida', 'Orlando'),

(43, 'amelia@example.com', 'Amelia', 'Gray', '555 Oak', '52639', 'Illinois', 'Springfield'),

(44, 'gavin@example.com', 'Gavin', 'Hudson', '666 Elm', '27635', 'Washington', 'Spokane'),

(45, 'isabella@example.com', 'Isabella', 'Collins', '777 Maple', '62539', 'Massachusetts', 'Worcester'),

(46, 'henry@example.com', 'Henry', 'Bryant', '888 Walnut', '94637', 'Colorado', 'Colorado Springs'),

(47, 'violet@example.com', 'Violet', 'Parker', '999 Pine', '47582', 'Arizona', 'Tucson'),

(48, 'caleb@example.com', 'Caleb', 'Murphy', '101 Cedar', '83746', 'Georgia', 'Savannah'),

(49, 'lillian@example.com', 'Lillian', 'Harris', '111 Oak', '36482', 'California', 'San Diego'),

(50, 'gabriel@example.com', 'Gabriel', 'Turner', '222 Elm', '59273', 'New York', 'Albany');

INSERT INTO CUSTOMER\_CONTACTNO (CONTACT\_NUMBER, CUSTOMER\_ID)

VALUES

('+92-300-1234567', 1),

('+92-321-9876543', 2),

('+92-333-5555555', 3),

('+92-345-8888888', 4),

('+92-300-9999999', 5),

('+92-300-1234666', 1),

('+92-321-9876577', 2),

('+92-333-5555444', 3),

('+92-345-8866888', 4),

('+92-300-9119999', 5),

('+92-300-1210567', 1),

('+92-321-9878543', 2),

('+92-333-5578955', 3),

('+92-345-8811888', 4),

('+92-300-9990999', 5);

INSERT INTO PAYMENT (PAYMENT\_ID, CUSTOMER\_ID, PAYMENT\_DATE, PAYMENT\_AMOUNT, PAYMENT\_METHOD) VALUES

(1, 50, '2023-05-01', 50.00, 'Credit Card'),

(2, 49, '2023-05-02', 20.00, 'PayPal'),

(3, 48, '2023-05-03', 15.99, 'Credit Card'),

(4, 47, '2023-05-04', 10.50, 'PayPal'),

(5, 46, '2023-05-05', 100.00, 'Credit Card'),

(6, 45, '2023-05-06', 75.50, 'Credit Card'),

(7, 44, '2023-05-07', 40.00, 'PayPal'),

(8, 43, '2023-05-08', 60.25, 'Credit Card'),

(9, 42, '2023-05-09', 12.99, 'PayPal'),

(10, 41, '2023-05-10', 200.00, 'Credit Card'),

(11, 39, '2023-05-11', 30.00, 'Credit Card'),

(12, 38, '2023-05-12', 18.75, 'PayPal'),

(13, 37, '2023-05-13', 100.99, 'Credit Card'),

(14, 36, '2023-05-14', 75.00, 'PayPal'),

(15, 35, '2023-05-15', 80.50, 'Credit Card'),

(16, 34, '2023-05-16', 45.00, 'PayPal'),

(17, 33, '2023-05-17', 150.25, 'Credit Card'),

(18, 32, '2023-05-18', 25.99, 'PayPal'),

(19, 31, '2023-05-19', 70.00, 'Credit Card'),

(20, 30, '2023-05-20', 90.50, 'Credit Card'),

(21, 29, '2023-05-21', 60.00, 'PayPal'),

(22, 28, '2023-05-22', 42.75, 'Credit Card'),

(23, 27, '2023-05-23', 80.99, 'PayPal'),

(24, 26, '2023-05-24', 55.00, 'Credit Card'),

(25, 25, '2023-05-25', 175.25, 'PayPal'),

(26, 24, '2023-05-26', 35.99, 'Credit Card'),

(27, 23, '2023-05-27', 90.00, 'PayPal'),

(28, 22, '2023-05-28', 120.50, 'Credit Card'),

(29, 21, '2023-05-29', 15.00, 'Credit Card'),

(30, 20, '2023-05-30', 50.75, 'PayPal');

INSERT INTO PRODUCT(PRODUCT\_ID, PRODUCT\_NAME, PRODUCT\_PRICE, PRODUCT\_DESCRIPTION, PRODUCTION\_YEAR, CATEGORY\_ID) VALUES

(1, 'iPhone 12', 999.00, 'Flagship smartphone from Apple','2008-11-11', 1),

(2, 'Samsung Galaxy S21', 899.00, 'High-end Android smartphone from Samsung','2021-09-09' , 2),

(3, 'Nike Air Max', 129.00, 'Popular athletic shoes with cushioning and style', '2021-1-17', 2),

(4, 'Levi''s Jeans', 79.00, 'Classic denim jeans from Levi''s', '2005-11-18', 2),

(5, 'KitchenAid Mixer', 349.00, 'Versatile kitchen mixer for baking and cooking', '2021-11-10', 3),

(6, 'Sony 65" 4K TV', 1499.00, 'Ultra-high-definition smart TV with stunning visuals', '2021-1-30', 1),

(7, 'Adidas Ultraboost', 169.00, 'Comfortable and stylish running shoes from Adidas', '2022-1-12', 2),

(8, 'Amazon Echo Dot', 49.99, 'Smart speaker with Alexa voice assistant', '2022-09-1', 3),

(9, 'Harry Potter and the Sorcerer''s Stone', 14.99, 'First book in the Harry Potter series by J.K. Rowling', '1997-03-22', 4),

(10, 'Wilson Tennis Racquet', 89.99, 'Professional-grade racquet for tennis players', '2022-10-21', 5),

(11, 'MacBook Pro', 1999.00, 'Powerful and sleek laptop for professionals', '2022-02-18', 5),

(12, 'Calvin Klein T-Shirt', 29.99, 'Classic and comfortable t-shirt from Calvin Klein','2018-10-11', 2),

(13, 'Cuisinart Blender', 79.00, 'High-performance blender for smoothies and more', '2000-2-12', 3),

(14, 'The Great Gatsby', 9.99, 'Classic novel by F. Scott Fitzgerald', '1995-10-21', 4),

(15, 'Yoga Mat', 19.99, 'Non-slip mat for yoga and fitness exercises', '2018-11-10', 5),

(16, 'Samsung 55" QLED TV', 1299.00, 'High-quality smart TV with vibrant colors', '2022-10-11', 1),

(17, 'Puma Sneakers', 89.99, 'Stylish and comfortable sneakers from Puma', '2019-11-11', 2),

(18, 'Instant Pot', 99.99, 'Multi-function electric pressure cooker and kitchen appliance','2020-1-29', 3),

(19, 'To Kill a Mockingbird', 12.99, 'Classic novel by Harper Lee', '1960-08-11', 5),

(20, 'Basketball', 24.99, 'Official size and weight basketball','1888-09-11', 5),

(21, 'Apple Watch Series 6', 399.00, 'Advanced smartwatch with health monitoring features', '2016-07-21', 1),

(22, 'Ralph Lauren Polo Shirt', 49.99, 'Stylish and comfortable polo shirt','2019-07-21' , 2),

(23, 'Keurig Coffee Maker', 129.00, 'Single-serve coffee maker with multiple brew sizes','2015-06-1', 3),

(24, 'The Lord of the Rings Trilogy', 29.99, 'Epic fantasy book series by J.R.R. Tolkien', '2016-05-21', 4),

(25, 'Outdoor Camping Tent', 79.99, 'Spacious tent for outdoor camping adventures','2016-07-22', 5),

(26, 'Google Pixel 5', 799.00, 'Android smartphone with exceptional camera capabilities','2015-07-21', 1),

(27, 'Under Armour Leggings', 59.99, 'Comfortable leggings for workouts and active wear','2014-11-21', 2),

(28, 'NutriBullet Blender', 69.00, 'Compact blender for making smoothies and shakes','2016-08-21', 3),

(29, '1984 by George Orwell', 11.99, 'Dystopian novel by George Orwell', '1994-09-21', 4),

(30, 'Dumbbell Set', 49.99, 'Adjustable dumbbells for strength training', '1991-07-16', 5),

(31, 'Sony PlayStation 5', 499.00, 'Next-generation gaming console from Sony','2007-12-21', 1),

(32, 'Tommy Hilfiger Hoodie', 79.99, 'Comfortable and stylish hoodie from Tommy Hilfiger', '2021-12-1', 2),

(33, 'Hamilton Beach Toaster', 39.99, 'Four-slice toaster with multiple settings','2016-02-22', 3),

(34, 'Ray-Ban Sunglasses', 129.99, 'Classic sunglasses with UV protection','2021-02-21', 2),

(35, 'Crock-Pot Slow Cooker', 49.99, 'Programmable slow cooker for convenient cooking', '1996-02-09', 3),

(36, 'The Catcher in the Rye', 8.99, 'Coming-of-age novel by J.D. Salinger','1959-07-21', 4),

(37, 'Soccer Ball', 14.99, 'Official size and weight soccer ball','1919-07-05', 5),

(38, 'Canon EOS Rebel T7', 599.00, 'Entry-level DSLR camera for photography enthusiasts', '1899-05-21', 1),

(39, 'Nike Air Force 1', 109.00, 'Iconic sneakers with timeless design', '2022-09-21', 2),

(40, 'Instant Camera', 79.99, 'Compact camera for instant photo printing','2002-02-28', 3),

(41, 'Amazon Echo Dot (4th Gen)', 49.99, 'Smart speaker with Alexa voice assistant','2016-07-10', 3),

(42, 'Fitbit Charge 4', 149.99, 'Fitness tracker with built-in GPS and heart rate monitoring','2007-07-07', 1),

(43, 'Adidas Stan Smith', 79.99, 'Iconic sneakers with clean and classic design', '2000-10-20', 2),

(44, 'Amazon Kindle Paperwhite', 129.99, 'Waterproof e-reader with built-in adjustable light','1898-11-12', 3),

(45, 'The Hobbit', 10.99, 'Fantasy novel by J.R.R. Tolkien', '1937-07-28', 4),

(46, 'Mountain Bike', 499.99, 'Durable and versatile bike for off-road adventures', '2010-05-29', 5),

(47, 'Bose QuietComfort 35 II', 299.00, 'Wireless noise-canceling headphones','1977-07-1', 1),

(48, 'Michael Kors Watch', 199.99, 'Elegant and stylish wristwatch', '1986-04-04', 2),

(49, 'Ninja Foodi Grill', 229.00, 'Indoor grill with multiple cooking functions','1998-03-10', 3),

(50, 'The Alchemist', 14.99, 'Bestselling novel by Paulo Coelho', '1998-02-02', 4),

(51, 'Resistance Bands Set', 29.99, 'Versatile workout equipment for strength training', '2001-01-01', 5);

INSERT INTO CART (CART\_ID, TOTAL\_COST, PRODUCT\_ID, TOTAL\_PRODUCT\_QUANTITY)

VALUES (1, 25.99, 51, 2),(2, 39.99, 12, 1),(3, 15.50, 3, 3),(4, 12.00, 4, 4),(5, 9.99, 22, 2),

(6, 49.99, 9, 1),(7, 29.99, 51, 2),(8, 19.99, 50, 3),(9, 7.99, 49, 1),(10, 14.50, 48, 2),

(11, 9.99, 47, 1),(12, 35.99, 46, 3),(13, 12.99, 45, 1),(14, 9.99, 44, 2),(15, 25.99, 43, 1),

(16, 19.99, 42, 1),(17, 39.99, 41, 2),(18, 14.50, 40, 1),(19, 9.99, 39, 2),(20, 49.99, 38, 1),

(21, 29.99, 37, 3),(22, 19.99, 36, 1),(23, 7.99, 35, 2),(24, 14.50, 34, 1),(25, 9.99, 33, 1),

(26, 35.99, 32, 2),(27, 12.99, 31, 1);

INSERT INTO COURIER (COURIER\_ID, COURIER\_NAME, EXPECTED\_DELIVERY\_DATE, CURRENT\_LOCATION, CART\_ID, CUSTOMER\_ID) VALUES

(1, 'DHL', '2023-06-10', 'New York', 1, 1),

(2, 'FedEx', '2023-06-12', 'Chicago', 2, 2),

(3, 'UPS', '2023-06-08', 'Los Angeles', 3, 3),

(4, 'USPS', '2023-06-09', 'Miami', 4, 4),

(5, 'TNT', '2023-06-11', 'Dallas', 5, 5),

(6, 'UPS', '2023-06-10', 'Seattle', 6, 6),

(7, 'DHL', '2023-06-11', 'San Francisco', 7, 7),

(8, 'FedEx', '2023-06-13', 'Houston', 8, 8),

(9, 'USPS', '2023-06-09', 'Boston', 9, 9),

(10, 'TNT', '2023-06-14', 'Atlanta', 10, 10),

(11, 'DHL', '2023-06-10', 'Chicago', 11, 11),

(12, 'UPS', '2023-06-12', 'New York', 12, 12),

(13, 'FedEx', '2023-06-15', 'Los Angeles', 13, 13),

(14, 'USPS', '2023-06-09', 'Seattle', 14, 14),

(15, 'TNT', '2023-06-11', 'San Francisco', 15, 15),

(16, 'UPS', '2023-06-12', 'Houston', 16, 16),

(17, 'DHL', '2023-06-13', 'Miami', 17, 17),

(18, 'FedEx', '2023-06-14', 'Boston', 18, 18),

(19, 'USPS', '2023-06-09', 'Atlanta', 19, 19),

(20, 'TNT', '2023-06-10', 'Chicago', 20, 20),

(21, 'DHL', '2023-06-11', 'New York', 21, 21),

(22, 'UPS', '2023-06-15', 'Los Angeles', 22, 22),

(23, 'FedEx', '2023-06-09', 'Seattle', 23, 23),

(24, 'USPS', '2023-06-12', 'San Francisco', 24, 24),

(25, 'TNT', '2023-06-13', 'Houston', 25, 25),

(26, 'UPS', '2023-06-14', 'Boston', 26, 26),

(27, 'DHL', '2023-06-09', 'Atlanta', 27, 27);

INSERT INTO COURIER\_CONTACTNO (CONTACT\_NO, COURIER\_ID)

VALUES

('0300-1234567', 1),

('0312-9876543', 2),

('0321-4567890', 3),

('0333-8765432', 4),

('0345-2345678', 5),

('0355-6789012', 6),

('0366-3456789', 7),

('0377-8901234', 8),

('0388-4567890', 9),

('0399-0987654', 10),

('0300-1111111', 11),

('0312-2222222', 12),

('0321-3333333', 13),

('0333-4444444', 14),

('0345-5555555', 15),

('0355-6666666', 16),

('0366-7777777', 17),

('0377-8888888', 18),

('0388-9999999', 19),

('0399-0000000', 20);

INSERT INTO COURIER\_CONTACTNO (CONTACT\_NO, COURIER\_ID)

VALUES

('0300-1234566', 1),

('0312-9876544', 2),

('0321-4567899', 3),

('0333-8765433', 4),

('0345-2345677', 5),

('0355-6789011', 6),

('0366-3456788', 7),

('0377-8901233', 8),

('0388-4567899', 9),

('0399-0987655', 10),

('0300-0111110', 11),

('0312-0222220', 12),

('0321-0333330', 13),

('0333-0444440', 14),

('0345-0555550', 15),

('0355-0666660', 16),

('0366-0777770', 17),

('0377-0888880', 18),

('0388-0999990', 19),

('0399-1000001', 20);

INSERT INTO DELIVERTO (DELIVERY\_ID, DELIVERY\_TYPE, CUSTOMER\_ID, COURIER\_ID) VALUES

(1, 'Standard', 1, 1),

(2, 'Express', 2, 2),

(3, 'Standard', 3, 3),

(4, 'Express', 4, 4),

(5, 'Standard', 5, 5),

(6, 'Standard', 6, 6),

(7, 'Express', 7, 7),

(8, 'Standard', 8, 8),

(9, 'Express', 9, 9),

(10, 'Standard', 10, 10),

(11, 'Standard', 11, 11),

(12, 'Express', 12, 12),

(13, 'Standard', 13, 13),

(14, 'Express', 14, 14),

(15, 'Standard', 15, 15),

(16, 'Standard', 16, 16),

(17, 'Express', 17, 17),

(18, 'Standard', 18, 18),

(19, 'Express', 19, 19),

(20, 'Standard', 20, 20),

(21, 'Standard', 21, 21),

(22, 'Express', 22, 22),

(23, 'Standard', 23, 23),

(24, 'Express', 24, 24),

(25, 'Standard', 25, 25),

(26, 'Standard', 26, 26),

(27, 'Express', 27, 27);

SQL Queries

**ALTER QURERY**

TRUNCATE TABLE CATEGORY

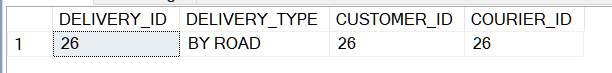
ALTER TABLE CATEGORY ADD CATEGORY\_DATE INT DEFAULT 0;

a

**UPDATE QURERY**

UPDATE DELIVERTO SET DELIVERY\_TYPE='BY ROAD' WHERE DELIVERY\_ID=26

SELECT \* FROM DELIVERTO WHERE DELIVERY\_ID = 26 AND DELIVERY\_TYPE = 'BY ROAD'



UPDATE CUSTOMER SET CUSTOMER\_EMAIL='gabriel@raybal.com' WHERE CUSTOMER\_ID=50

SELECT \* FROM CUSTOMER WHERE CUSTOMER\_EMAIL='gabriel@raybal.com';



DELETE FROM DELIVERTO WHERE DELIVERY\_ID=27

**OPERATORS**

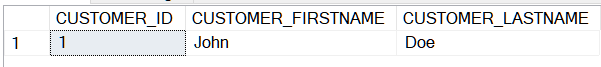
SELECT CUSTOMER\_FIRSTNAME FROM CUSTOMER WHERE CUSTOMER\_FIRSTNAME LIKE '%a%' OR CUSTOMER\_FIRSTNAME LIKE 'D%'





SELECT CUSTOMER\_ID,CUSTOMER\_FIRSTNAME,CUSTOMER\_LASTNAME FROM CUSTOMER

WHERE CUSTOMER\_FIRSTNAME='John' AND CUSTOMER\_LASTNAME ='Doe'

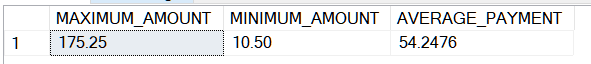


**AGGREGATE FUNCTION**

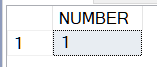
SELECT MAX(PAYMENT\_AMOUNT) AS MAXIMUM\_AMOUNT,MIN(PAYMENT\_AMOUNT) AS MINIMUM\_AMOUNT,AVG(PAYMENT\_AMOUNT) AS AVERAGE\_PAYMENT

FROM PAYMENT

WHERE PAYMENT\_METHOD LIKE 'P%'



SELECT COUNT(\*) AS NUMBER FROM CUSTOMER WHERE CUSTOMER\_EMAIL LIKE '%raybal%'

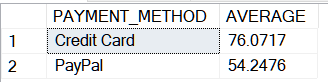


**GROUP BY**

SELECT PAYMENT\_METHOD,AVG(PAYMENT\_AMOUNT) AS AVERAGE

FROM PAYMENT

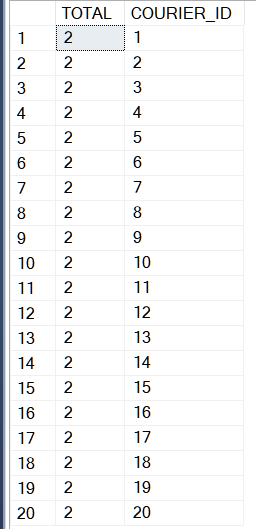
GROUP BY(PAYMENT\_METHOD)



SELECT COUNT(\*) AS TOTAL,COURIER\_ID

FROM COURIER\_CONTACTNO

GROUP BY(COURIER\_ID)

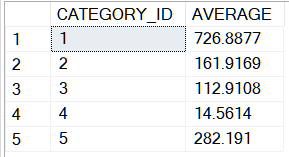


**/\*HAVING - Used to filter out aggregate functions\*/**

SELECT CATEGORY\_ID,AVG(PRODUCT\_PRICE) AS AVERAGE

FROM PRODUCT

GROUP BY(CATEGORY\_ID)



SELECT CATEGORY\_ID,AVG(PRODUCT\_PRICE) AS AVERAGE

FROM PRODUCT

GROUP BY(CATEGORY\_ID)

HAVING AVG(PRODUCT\_PRICE)>700.00

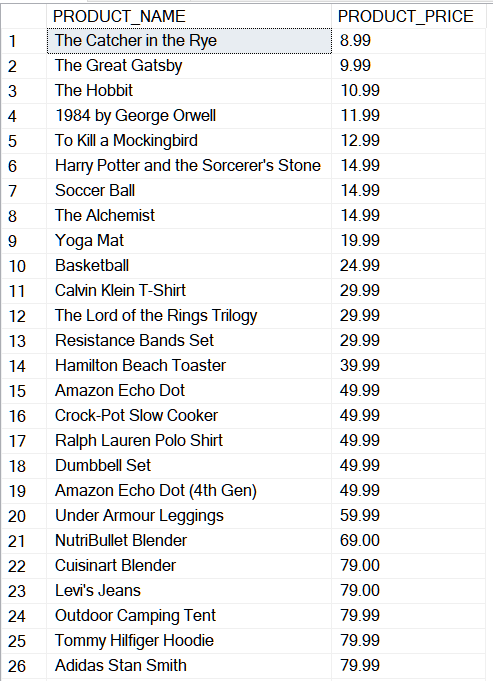


**/\*ORDER BY\*/**

SELECT PRODUCT\_NAME,PRODUCT\_PRICE

FROM PRODUCT

ORDER BY PRODUCT\_PRICE ASC





SELECT PRODUCT\_NAME,PRODUCT\_PRICE

FROM PRODUCT

ORDER BY PRODUCT\_PRICE DESC

**JOINS**

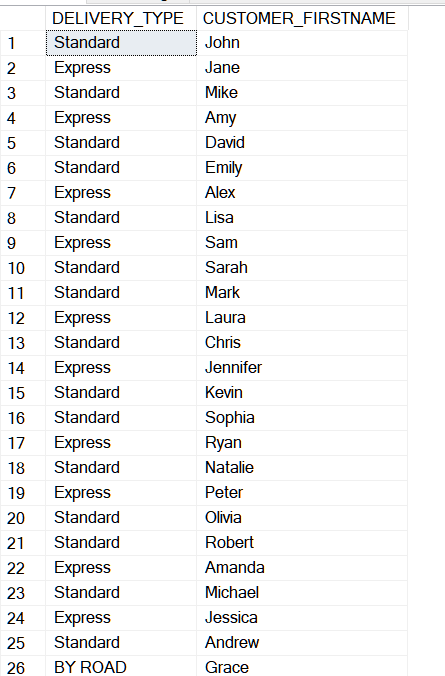
SELECT D.DELIVERY\_TYPE,C.CUSTOMER\_FIRSTNAME

FROM DELIVERTO D

INNER JOIN

CUSTOMER C

ON C.CUSTOMER\_ID = D.CUSTOMER\_ID



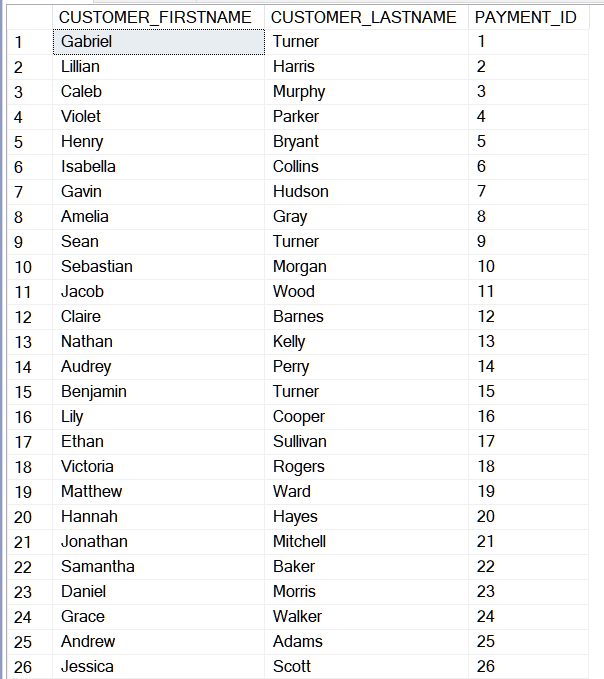
SELECT C.CUSTOMER\_FIRSTNAME,C.CUSTOMER\_LASTNAME,P.PAYMENT\_ID

FROM PAYMENT P

LEFT JOIN

CUSTOMER C

ON P.CUSTOMER\_ID = C.CUSTOMER\_ID





SELECT C.CUSTOMER\_FIRSTNAME,C.CUSTOMER\_LASTNAME,P.PAYMENT\_ID

FROM PAYMENT P

RIGHT JOIN

CUSTOMER C

ON P.CUSTOMER\_ID = C.CUSTOMER\_ID

