#### **Opis wymagań**

The database will concern a motorcycle parts store and will store information about customers, orders, products, suppliers and payments. Below is a detailed description of the requirements:

1. **Klienci (Customers)**:

* Customer information such as name, surname, address, telephone number and email.

1. **Produkty (Products)**:

* Details about products available in the store, such as product name, description, price, stock and supplier.

1. **Dostawcy (Suppliers)**:

* Supplier information such as company name, contact, address, telephone number and email.

1. **Zamówienia (Orders)**:

* Details about orders such as order date, customer, order status and total amount.

1. **Pozycje zamówienia (OrderItems)**:

* Information about individual order items, such as product, quantity, unit price, and total amount for the item.

1. **Płatności (Payments)**:

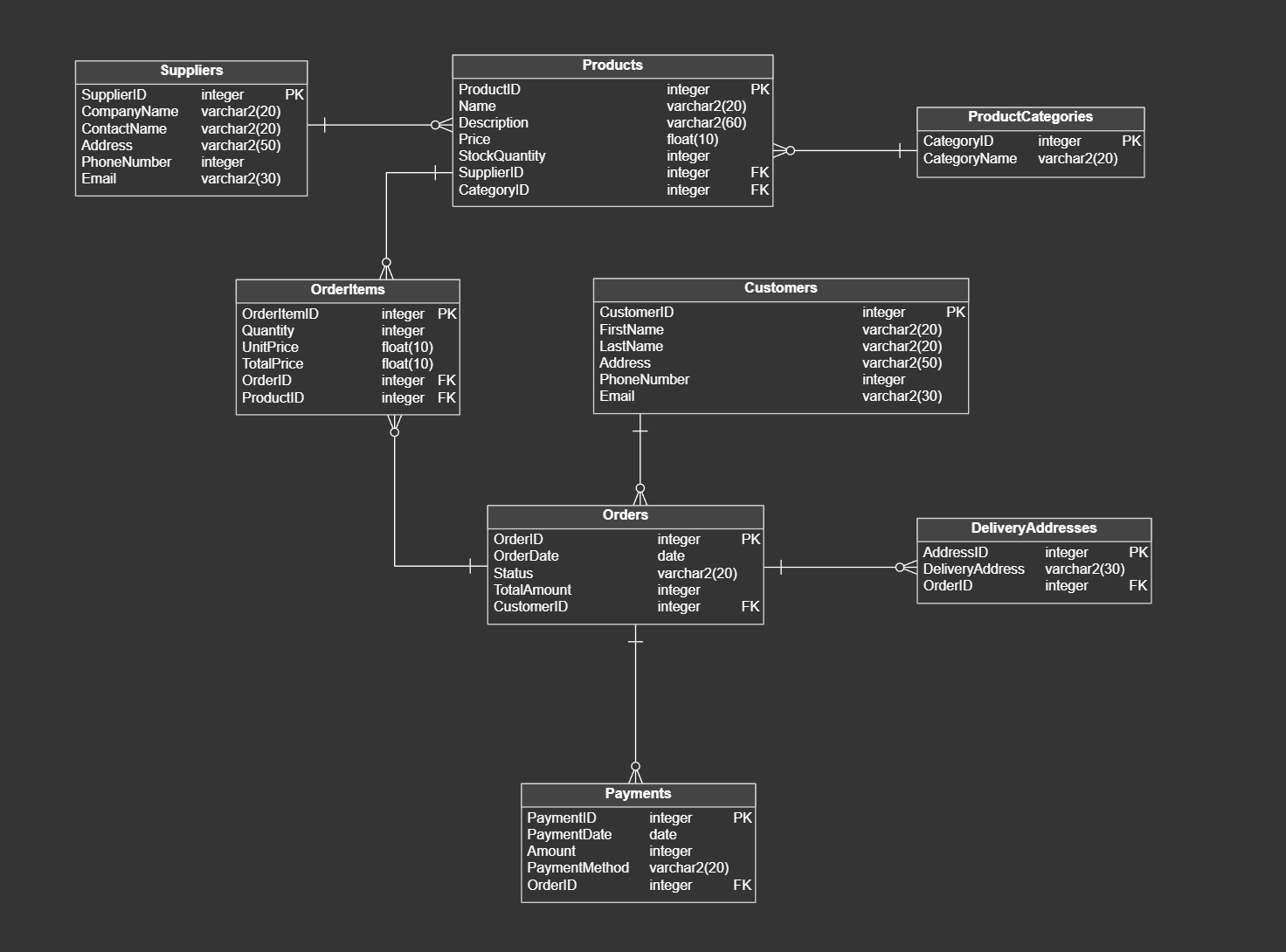
* Payment information such as payment date, amount, payment method and associated order.

1. **Kategorie produktów (ProductCategories)**:

* Product categories such as engines, tires, brakes, etc.

1. **Adresy dostaw (DeliveryAddresses)**:

* Delivery addresses for customer orders.



***Creating tables!***

-- Created by Vertabelo (http://vertabelo.com)

-- tables

-- Table: Customers

CREATE TABLE Customers (

CustomerID integer NOT NULL,

FirstName varchar2(20) NOT NULL,

LastName varchar2(20) NOT NULL,

Address varchar2(50) NOT NULL,

PhoneNumber integer NOT NULL,

Email varchar2(30) NOT NULL,

CONSTRAINT Customers\_pk PRIMARY KEY (CustomerID)

) ;

-- Table: DeliveryAddresses

CREATE TABLE DeliveryAddresses (

AddressID integer NOT NULL,

DeliveryAddress varchar2(30) NOT NULL,

OrderID integer NOT NULL,

CONSTRAINT DeliveryAddresses\_pk PRIMARY KEY (AddressID)

) ;

-- Table: OrderItems

CREATE TABLE OrderItems (

OrderItemID integer NOT NULL,

Quantity integer NOT NULL,

UnitPrice float(10) NOT NULL,

TotalPrice float(10) NOT NULL,

OrderID integer NOT NULL,

ProductID integer NOT NULL,

CONSTRAINT OrderItems\_pk PRIMARY KEY (OrderItemID)

) ;

-- Table: Orders

CREATE TABLE Orders (

OrderID integer NOT NULL,

OrderDate date NOT NULL,

Status varchar2(20) NOT NULL,

TotalAmount integer NOT NULL,

CustomerID integer NOT NULL,

CONSTRAINT Orders\_pk PRIMARY KEY (OrderID)

) ;

-- Table: Payments

CREATE TABLE Payments (

PaymentID integer NOT NULL,

PaymentDate date NOT NULL,

Amount integer NOT NULL,

PaymentMethod varchar2(20) NOT NULL,

OrderID integer NOT NULL,

CONSTRAINT Payments\_pk PRIMARY KEY (PaymentID)

) ;

-- Table: ProductCategories

CREATE TABLE ProductCategories (

CategoryID integer NOT NULL,

CategoryName varchar2(20) NOT NULL,

CONSTRAINT ProductCategories\_pk PRIMARY KEY (CategoryID)

) ;

-- Table: Products

CREATE TABLE Products (

ProductID integer NOT NULL,

Name varchar2(20) NOT NULL,

Description varchar2(60) NOT NULL,

Price float(10) NOT NULL,

StockQuantity integer NOT NULL,

SupplierID integer NOT NULL,

CategoryID integer NOT NULL,

CONSTRAINT Products\_pk PRIMARY KEY (ProductID)

) ;

-- Table: Suppliers

CREATE TABLE Suppliers (

SupplierID integer NOT NULL,

CompanyName varchar2(20) NOT NULL,

ContactName varchar2(20) NOT NULL,

Address varchar2(50) NOT NULL,

PhoneNumber integer NOT NULL,

Email varchar2(30) NOT NULL,

CONSTRAINT Suppliers\_pk PRIMARY KEY (SupplierID)

) ;

-- foreign keys

-- Reference: DeliveryAddresses\_Orders (table: DeliveryAddresses)

ALTER TABLE DeliveryAddresses ADD CONSTRAINT DeliveryAddresses\_Orders

FOREIGN KEY (OrderID)

REFERENCES Orders (OrderID);

-- Reference: OrderItems\_Orders (table: OrderItems)

ALTER TABLE OrderItems ADD CONSTRAINT OrderItems\_Orders

FOREIGN KEY (OrderID)

REFERENCES Orders (OrderID);

-- Reference: OrderItems\_Products (table: OrderItems)

ALTER TABLE OrderItems ADD CONSTRAINT OrderItems\_Products

FOREIGN KEY (ProductID)

REFERENCES Products (ProductID);

-- Reference: Orders\_Customers (table: Orders)

ALTER TABLE Orders ADD CONSTRAINT Orders\_Customers

FOREIGN KEY (CustomerID)

REFERENCES Customers (CustomerID);

-- Reference: Payments\_Orders (table: Payments)

ALTER TABLE Payments ADD CONSTRAINT Payments\_Orders

FOREIGN KEY (OrderID)

REFERENCES Orders (OrderID);

-- Reference: Products\_ProductCategories (table: Products)

ALTER TABLE Products ADD CONSTRAINT Products\_ProductCategories

FOREIGN KEY (CategoryID)

REFERENCES ProductCategories (CategoryID);

-- Reference: Products\_Suppliers (table: Products)

ALTER TABLE Products ADD CONSTRAINT Products\_Suppliers

FOREIGN KEY (SupplierID)

REFERENCES Suppliers (SupplierID);

-- End of file.

***Inserts !***

INSERT INTO Customers (customerid,FirstName, LastName, Address, PhoneNumber, Email) VALUES

(1,'Anna', 'Nowak', 'ul. Słoneczna 5', '987654321', 'anna.nowak@example.com')

—--

(2,'Jan', 'Kowalski', 'ul. Główna 10', '123456789', '[jan.kowalski@example.com](mailto:jan.kowalski@example.com)');

—--

('3','Dawid', 'Juszynski', 'ul. Koszykowa', '123456789', '[jan.kowalski@example.com](mailto:jan.kowalski@example.com)')

INSERT INTO Suppliers (supplierid,CompanyName, ContactName, Address, PhoneNumber, Email) VALUES

(1,'Motor Parts Co.', 'Pawel Szla', 'ul. Fabryczna 33', '234562290', 'contact@motorparts.com')

—---

(2,'Moto Accessories', 'Piotr Miazga', 'ul. Rowwerowa 24', '345678901', 'contact@bikeaccessories.com');

INSERT INTO ProductCategories (categoryid, CategoryName) VALUES

(1, 'Silniki')

—---

(2, 'Opony')

—---

(3, 'Hamulec')

INSERT INTO Products (PRODUCTID,Name, Description, Price, StockQuantity, SupplierID, CategoryID) VALUES

(1,'Silnik 500cc', 'Silnik o pojemności 500cc', 2500.00, 10, 1, 1)

—---

(2,'Opona sportowa', 'Opona przeznaczona do sportowych motocykli', 300.00, 50, 2, 2)

—---

(3,'Klocki hamulcowe', 'Klocki hamulcowe do różnych modeli motocykli', 150.00, 30, 1, 3)

INSERT INTO Orders (orderid, OrderDate, CustomerID, Status, TotalAmount) VALUES

(1,'2023-01-15', 1, 'Zrealizowane', 2800.00)

—---

(2,'2023-02-20', 2, 'W trakcie realizacji', 450.00)

INSERT INTO OrderItems (orderitemid, OrderID, ProductID, Quantity, UnitPrice, TotalPrice) VALUES

(1,1, 1, 1, 2500.00, 2500.00)

—---

(2,1, 2, 1, 300.00, 300.00)

—---

(3,2, 3, 3, 150.00, 450.00);

INSERT INTO Payments (paymentid, PaymentDate, Amount, PaymentMethod, OrderID) VALUES

(1,'2023-01-16', 2800.00, 'Karta kredytowa', 1)

—---

(2,'2024-01-27', 500.00, 'Przelew bankowy', 2);

INSERT INTO DeliveryAddresses (addressid, OrderID, DeliveryAddress) VALUES

(1,1, 'ul. Główna 10')

—---

(2,2, 'ul. Koszykowa 12')

***Inquiries!***

***Lista wszystkich klientów:***

***List of all customers:***

SELECT \* FROM Customers;

***Szczegóły zamówień dla klienta o ID 1.***

***Order details for customer ID 1.***

SELECT \* FROM Orders WHERE CustomerID = 1;

***Lista produktów wraz z kategoriami i dostawcami.***

***List of products with categories and suppliers.***

SELECT Products.Name, Products.Description, Products.Price, ProductCategories.CategoryName, Suppliers.CompanyName FROM Products JOIN ProductCategories ON Products.CategoryID = ProductCategories.CategoryID JOIN Suppliers ON Products.SupplierID = Suppliers.SupplierID;

***Suma zamówień dla każdego klienta***

***Total orders for each customer***

SELECT Customers.FirstName, Customers.LastName, SUM(Orders.TotalAmount) AS TotalSpent

FROM Customers

JOIN Orders ON Customers.CustomerID = Orders.CustomerID

GROUP BY Customers.CustomerID, Customers.FirstName, Customers.LastName;

***Lista produktów, które są na stanie w ilości mniejszej niż 11.***

***List of products that are in stock in quantities of less than 11.***

SELECT \* FROM Products WHERE StockQuantity <= 10;

***Szczegóły płatności dla zamówienia o ID 2.***

***Payment details for order ID 2.***

SELECT \* FROM Payments WHERE OrderID = 2;

***Lista zamówień wraz z produktami i ich ilością.***

***List of orders with products and their quantity.***

SELECT Orders.OrderID, Orders.OrderDate, Products.Name, OrderItems.Quantity

FROM Orders

JOIN OrderItems ON Orders.OrderID = OrderItems.OrderID

JOIN Products ON OrderItems.ProductID = Products.ProductID;

***Adresy dostaw dla wszystkich zamówień.***

***Delivery addresses for all orders.***

SELECT Orders.OrderID, DeliveryAddresses.DeliveryAddress FROM Orders JOIN DeliveryAddresses ON Orders.OrderID = DeliveryAddresses.OrderID;

***Łączna liczba produktów zamówionych przez klienta o ID 2.***

***Total number of products ordered by customer ID 2.***

SELECT SUM(OrderItems.Quantity) AS TotalProducts

FROM Orders

JOIN OrderItems ON Orders.OrderID = OrderItems.OrderID

WHERE Orders.CustomerID = 2;

***Szczegóły zamówień, które są w trakcie realizacji.***

***Details of orders that are in progress.***

SELECT \* FROM Orders WHERE Status = 'W trakcie realizacji';