PROJECT CATASCHEVASTICA

An Online Transaction Database (OLTP)

TEAM 5

Giarmas Nikos

Kakalis Loukas

Michailidis Giorgos

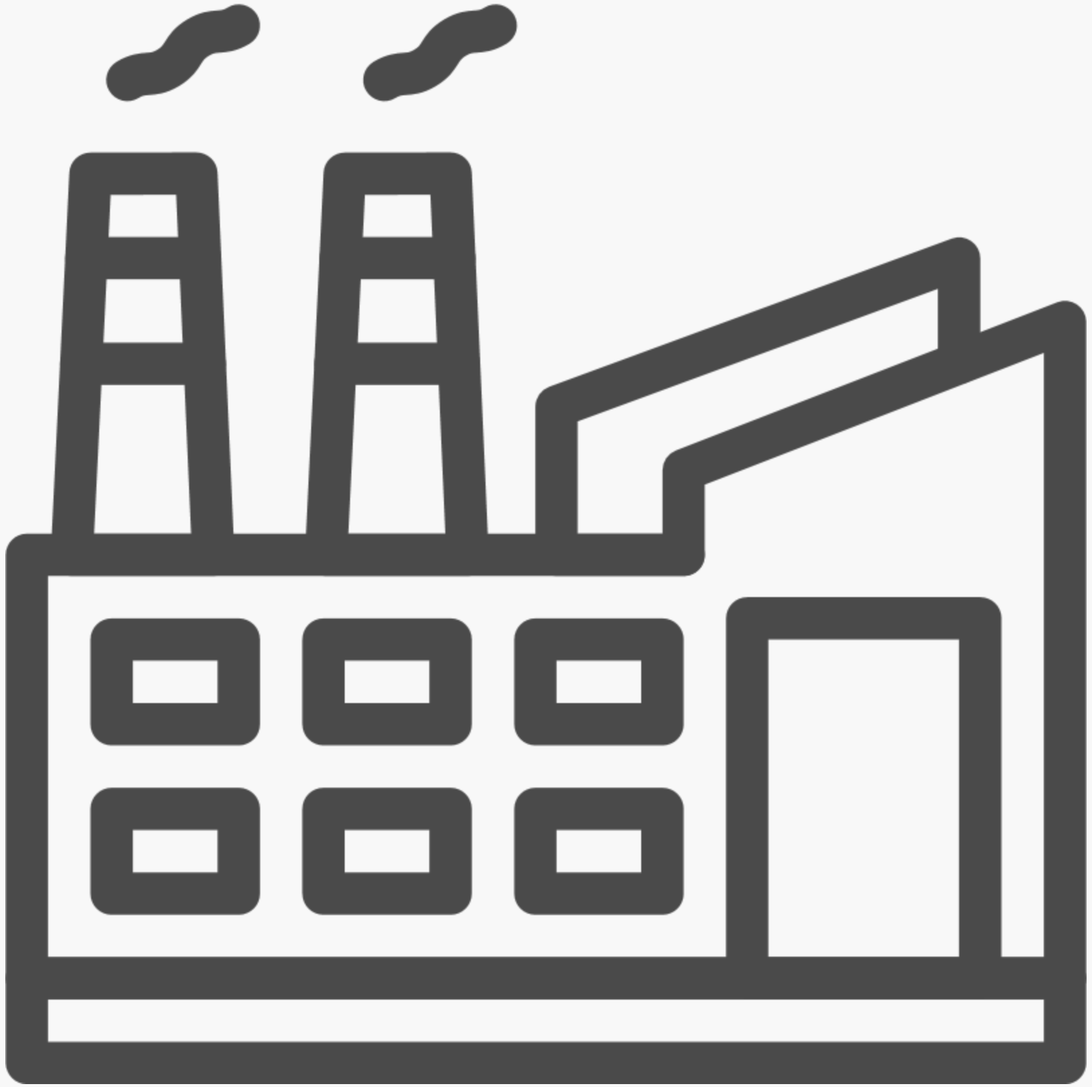
Mistiloglou Charis

Tzimi Katerina

Cataschevastica

Table of contents

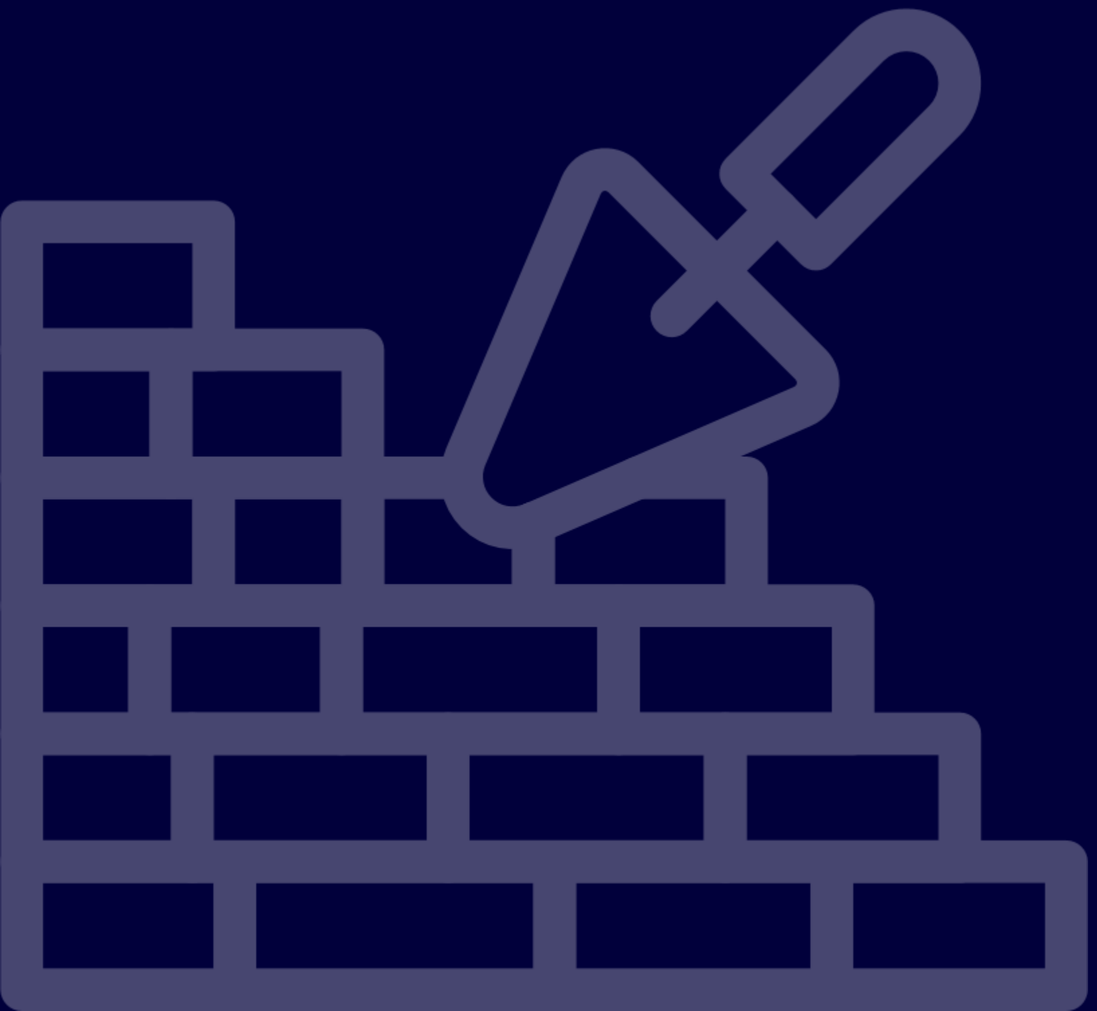
Cataschevastica



Executive Summary

- This project involves developing an Online Transaction Processing (OLTP) database for a fictional manufacturing company, Cataschevastica. The goal of this project is the development of a database that will manage the entire manufacturing process, from order intake and production to delivery, emphasizing customer satisfaction, operational efficiency, and service quality.

Project Development Roles



Business Analysts

- *Giarmas Nikos*
- *Michailidis Georgios*

Data Engineers

- *Mistiloglou Charalampos*
- *Tzimi Katerina*

Project Manager

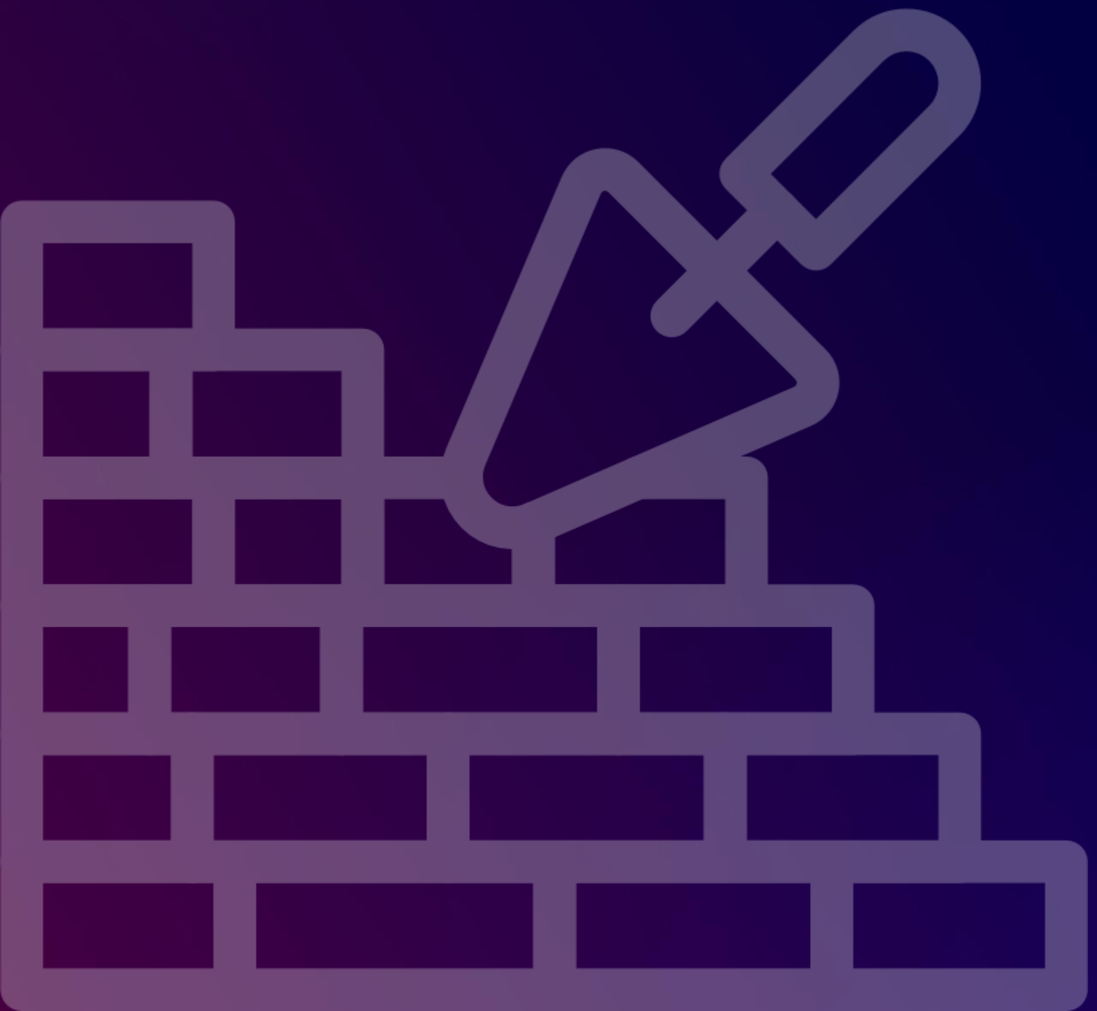
- *Kakalis Loukas*

Product Owner

- *Instructor*

Cataschevastica

Key Points

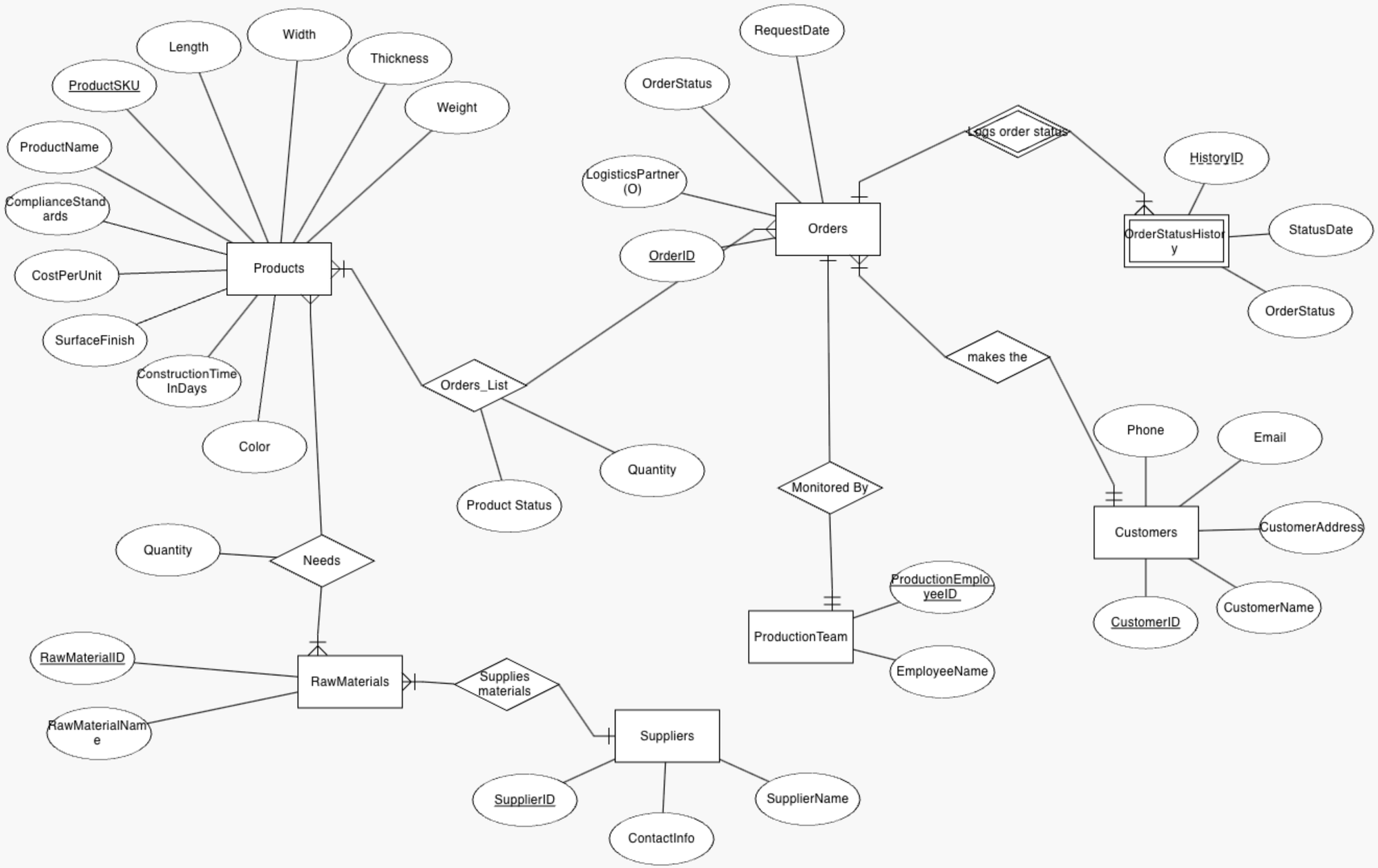


1. Products construction
- Create Product table
- Create Raw materials’ table as they are essential for finished product’s construction
- Create Suppliers table providing Raw materials

2. Orders procedure and delivery
- Create Orders table
- Create Customers table
- Create Production team table which monitors the orders state
- Create Order history table to log each order’s state over time
- Create Order list table where all products of an order are contained

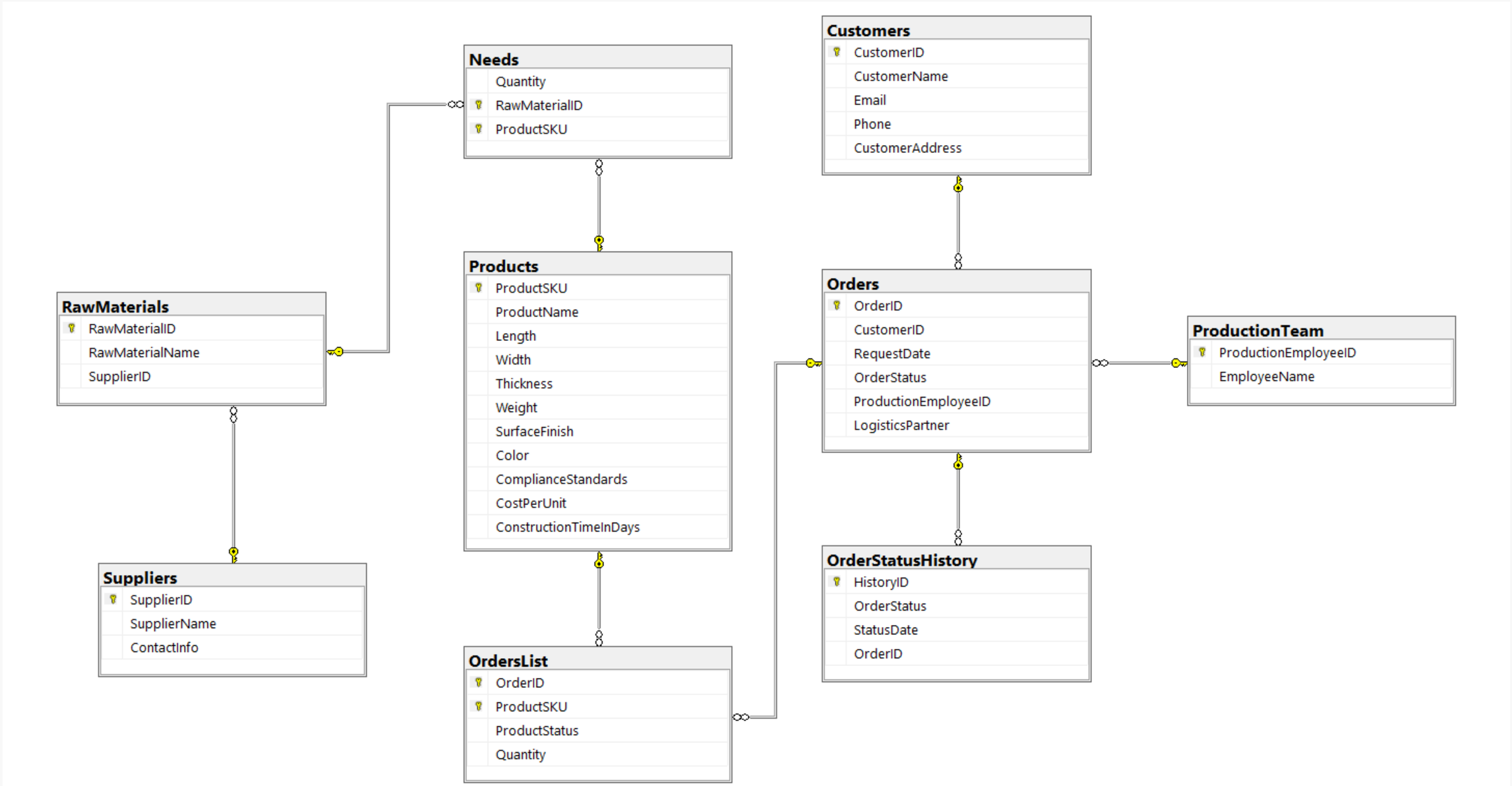
Cataschevastica

ERD Diagram



Cataschevastica

SQL Relational Schema



SQL - Creating database, tables

1st step - Creating the Database Cataschevastica

```
-- Create Cataschevastica Database
CREATE DATABASE Cataskevasticha;
GO
```

2nd step - Creating all the tables according to Entity Relationship Diagram

```
-- Orders Table
CREATE TABLE Orders (
    OrderID INT IDENTITY(1,1) PRIMARY KEY,
    CustomerID INT NOT NULL,
    RequestDate DATETIME,
    OrderStatus VARCHAR(100) NOT NULL DEFAULT 'In Process' CHECK (OrderStatus IN ('In Process', 'In Delivery', 'Complete', 'Cancelled'),
    ProductionEmployeeID INT,
    LogisticsPartner VARCHAR(100),
    CONSTRAINT FK_CustomerID FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    CONSTRAINT FK_ProductionEmployeeID FOREIGN KEY (ProductionEmployeeID) REFERENCES ProductionTeam(ProductionEmployeeID)
);
```

```
-- RawMaterials Table
CREATE TABLE RawMaterials (
    RawMaterialID INT IDENTITY(1,1) PRIMARY KEY,
    RawMaterialName VARCHAR(100),
    SupplierID INT,
    CONSTRAINT FK_SupplierID FOREIGN KEY (SupplierID) REFERENCES Suppliers(SupplierID)
);
```

```
-- Customers Table
CREATE TABLE Customers (
    CustomerID INT IDENTITY(1,1) PRIMARY KEY,
    CustomerName VARCHAR(100) NOT NULL,
    Email VARCHAR(255),
    Phone VARCHAR(20),
    CustomerAddress NVARCHAR(255)
);
```

```
-- OrderProductList Table
CREATE TABLE OrdersList (
    OrderID INT,
    ProductSKU VARCHAR(50),
    ProductStatus VARCHAR(100) DEFAULT 'In Production',
    Quantity INT NOT NULL,
    PRIMARY KEY(OrderID, ProductSKU),
    CONSTRAINT FK_OrderID FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    CONSTRAINT FK_ProductSKU FOREIGN KEY (ProductSKU) REFERENCES Products(ProductSKU)
);
```

```
-- Products Table
CREATE TABLE Products (
    ProductSKU VARCHAR(50) PRIMARY KEY,
    ProductName VARCHAR(100) NOT NULL,
    Length DECIMAL(10, 2) NOT NULL,
    Width DECIMAL(10, 2) NOT NULL,
    Thickness DECIMAL(10, 2) NOT NULL,
    Weight DECIMAL(10, 2) NOT NULL,
    SurfaceFinish NVARCHAR(255),
    Color VARCHAR(100),
    ComplianceStandards VARCHAR(100) NOT NULL,
    CostPerUnit DECIMAL(10, 2) NOT NULL,
    ConstructionTimeInDays INT NOT NULL
);
```

SQL - Populating tables

3rd Step - Populating the tables with data, 15 rows per table

```sql
-- Insert data into Products table
INSERT INTO Products (ProductSKU, ProductName, Length, Width, Thickness, Weight, SurfaceFinish, Color, ComplianceStandards, CostPerUnit, ConstructionTimeInDays)
VALUES
('BRK123', 'Brick', 10.00, 5.00, 0.50, 2.00, 'Matte', 'Red', 'ISO 9001', 15.00, 7),
('BLK456', 'Block', 12.00, 6.00, 0.75, 3.00, 'Glossy', 'Blue', 'ASTM C90', 20.00, 5),
('CON789', 'Concrete', 8.00, 4.00, 0.60, 1.50, 'Matte', 'Gray', 'ISO 14001', 10.00, 6),
('CEM321', 'Cement', 15.00, 7.00, 1.00, 4.00, 'Glossy', 'Gray', 'ASTM C150', 25.00, 8),
('STL654', 'Steel', 11.00, 5.50, 0.65, 2.20, 'Matte', 'Silver', 'ISO 9001', 18.00, 7),
('MET987', 'Metal Product', 9.00, 4.50, 0.70, 1.80, 'Glossy', 'Gold', 'ISO 14001', 14.00, 5),
('ROF543', 'Roofing', 13.00, 6.50, 0.85, 2.50, 'Matte', 'Brown', 'ASTM D7158', 22.00, 6),
('INS876', 'Insulation Material', 10.50, 5.25, 0.55, 2.10, 'Glossy', 'White', 'ASTM C518', 16.00, 8),
('TIL234', 'Tile', 12.50, 6.25, 0.80, 3.10, 'Matte', 'Blue', 'ISO 9001', 19.00, 7),
('BRD567', 'Board', 14.00, 7.00, 1.10, 4.20, 'Glossy', 'White', 'ASTM C208', 23.00, 5),
('GLS890', 'Glass', 11.50, 5.75, 0.90, 2.30, 'Matte', 'Transparent', 'ISO 14001', 17.00, 6),
('PLY012', 'Plywood', 9.50, 4.75, 0.75, 1.90, 'Glossy', 'Brown', 'ASTM D3043', 13.00, 7),
('PLB321', 'Plasterboard', 10.00, 5.00, 0.50, 1.50, 'Matte', 'White', 'EN 520', 12.00, 5),
('ASB654', 'Asbestos Cement', 11.00, 6.00, 0.75, 3.50, 'Glossy', 'Gray', 'ISO 9001', 22.00, 6),
('PVC987', 'PVC Panel', 9.00, 4.50, 0.70, 1.20, 'Glossy', 'White', 'ASTM D1784', 14.00, 4);
```

```sql
-- Insert data into Suppliers table
INSERT INTO Suppliers (SupplierName, ContactInfo)
VALUES
('AlphaBeta Supplies', 'contact@absupplies.com'),
('Edit Materials', 'info@editmaterials.com'),
('Quality Builders', 'support@qualitybuilders.com'),
('Delta Industrial', 'sales@deltaindustrial.com'),
('Pro Build', 'info@probuild.com'),
('Elite Construction', 'contact@eliteconstruction.com'),
('Top Grade Supplies', 'support@topgradesupplies.com'),
('BuildTech', 'info@buildtech.com'),
('Prime Materials', 'contact@primematerials.com'),
('HighTech Supplies', 'support@hightechsupplies.com'),
('MegaBuild', 'info@megabuild.com'),
('SuperBuild', 'contact@superbuild.com'),
('UltraMaterials', 'support@ultramaterials.com'),
('EcoBuild', 'info@ecobuild.com'),
('GreenBuild', 'contact@greenbuild.com');
```

```sql
-- Insert data into Orders table
INSERT INTO Orders (CustomerID, RequestDate, OrderStatus, ProductionEmployeeID, LogisticsPartner)
VALUES
(1, '2024-01-15', 'In Process', 1, 'FedEx'),
(2, '2024-02-20', 'In Process', 2, 'UPS'),
(3, '2024-03-10', 'In Process', 3, 'DHL'),
(4, '2024-04-25', 'In Process', 4, 'TNT'),
(5, '2024-05-15', 'In Process', 5, 'FedEx'),
(6, '2024-06-10', 'In Process', 6, 'UPS'),
(7, '2024-07-20', 'In Process', 7, 'DHL'),
(8, '2024-08-15', 'In Process', 8, 'TNT'),
(9, '2024-09-10', 'In Process', 9, 'FedEx'),
(10, '2024-10-05', 'In Process', 10, 'UPS'),
(11, '2024-11-15', 'In Process', 11, 'DHL'),
(12, '2024-12-10', 'In Process', 12, 'TNT'),
(13, '2025-01-15', 'In Process', 13, 'FedEx'),
(14, '2025-02-20', 'In Process', 14, 'UPS'),
(15, '2025-03-10', 'In Process', 15, 'DHL');
```

```sql
-- Insert data into Customers table
INSERT INTO Customers (CustomerName, Email, Phone, CustomerAddress)
VALUES
('John Doe', 'john.doe@example.com', '123-456-7890', '123 Main St, City A'),
('Jane Smith', 'jane.smith@example.com', '234-567-8901', '456 Oak St, City B'),
('Mike Johnson', 'mike.johnson@example.com', '345-678-9012', '789 Pine St, City C'),
('Emily Davis', 'emily.davis@example.com', '456-789-0123', '101 Maple St, City D'),
('Chris Brown', 'chris.brown@example.com', '567-890-1234', '202 Cedar St, City E'),
('Patricia Miller', 'patricia.miller@example.com', '678-901-2345', '303 Birch St, City F'),
('Robert Wilson', 'robert.wilson@example.com', '789-012-3456', '404 Elm St, City G'),
('Linda Martinez', 'linda.martinez@example.com', '890-123-4567', '505 Ash St, City H'),
('Barbara Anderson', 'barbara.anderson@example.com', '901-234-5678', '606 Walnut St, City I'),
('Michael Thomas', 'michael.thomas@example.com', '012-345-6789', '707 Chestnut St, City J'),
('Elizabeth Jackson', 'elizabeth.jackson@example.com', '123-456-7891', '808 Spruce St, City K'),
('David White', 'david.white@example.com', '234-567-8902', '909 Cypress St, City L'),
('Susan Harris', 'susan.harris@example.com', '345-678-9013', '1010 Sycamore St, City M'),
('James Clark', 'james.clark@example.com', '456-789-0124', '1111 Beech St, City N'),
('Mary Lewis', 'mary.lewis@example.com', '567-890-1235', '1212 Redwood St, City O');
```

```sql
-- Insert data into RawMaterials table
INSERT INTO RawMaterials (RawMaterialName, SupplierID)
VALUES
('Clay', 1),
('Cement Powder', 2),
('Sand', 1),
('Gravel', 4),
('Steel Rods', 5),
('Insulation Foam', 6),
('Wood', 7),
('Glass Fiber', 8),
('Plastic', 7),
('Aluminum', 10),
('Copper', 11),
('Rubber', 12),
('Gypsum', 13),
('Asbestos', 13),
('PVC', 15);
```

SQL Statements

4th Step - Quering the data

```sql
-- a. List of all products ordered yesterday (so that production can start):
SELECT
    op.ProductSKU,
    p.ProductName,
    op.ProductStatus,
    op.Quantity,
    o.RequestDate
FROM OrdersList op
JOIN Orders o ON op.OrderID = o.OrderID
JOIN Products p ON op.ProductSKU = p.ProductSKU
WHERE o.RequestDate = DATEADD(DAY, -1, CONVERT(DATE, GETDATE()));

-- b2. List of all finished orders ready to deliver:
SELECT
    OrderID,
    OrderStatus,
    RequestDate,
    LogisticsPartner
FROM Orders
WHERE OrderStatus = 'In Delivery';

-- c. List of all orders per customer, completed, pending, cancelled:
SELECT
    c.CustomerName,
    SUM(CASE WHEN o.OrderStatus = 'Complete' THEN 1 ELSE 0 END) AS CompletedOrders,
    SUM(CASE WHEN o.OrderStatus in ('In Process','In Delivery')  THEN 1 ELSE 0 END) AS PendingOrders,
    SUM(CASE WHEN o.OrderStatus = 'Cancelled' THEN 1 ELSE 0 END) AS CancelledOrders
FROM Orders o
LEFT JOIN Customers c ON c.CustomerID = o.CustomerID
GROUP BY
    c.CustomerName;
```

```sql
-- d. List of all products with quantities, ordered and delivered, ordered and pending, cancelled:
SELECT
    p.ProductSKU,
    p.ProductName,
    SUM(CASE WHEN op.ProductStatus = 'In Production' THEN op.Quantity ELSE 0 END) AS OrderedAndPending,
    SUM(CASE WHEN o.OrderStatus = 'Complete' THEN op.Quantity ELSE 0 END) AS OrderedAndDelivered,
    SUM(CASE WHEN o.OrderStatus = 'Cancelled' THEN op.Quantity ELSE 0 END) AS Cancelled
FROM Products p
LEFT JOIN OrdersList op ON p.ProductSKU = op.ProductSKU
LEFT JOIN Orders o ON op.OrderID = o.OrderID
GROUP BY
    p.ProductSKU,
    p.ProductName;


-- e. List of orders per production team employee, completed, pending, cancelled:
SELECT
    pt.EmployeeName,
    SUM(CASE WHEN o.OrderStatus = 'Complete' THEN 1 ELSE 0 END) AS CompletedOrders,
    SUM(CASE WHEN o.OrderStatus = 'In Process' THEN 1 ELSE 0 END) AS PendingOrders,
    SUM(CASE WHEN o.OrderStatus = 'Cancelled' THEN 1 ELSE 0 END) AS CancelledOrders
FROM ProductionTeam pt
LEFT JOIN Orders o ON pt.ProductionEmployeeID = o.ProductionEmployeeID
GROUP BY
    pt.EmployeeName;


-- f. Daily order and production report:
SELECT
    CONVERT(DATE, o.RequestDate) AS OrderDate,
    COUNT(o.OrderID) AS TotalOrders,
    SUM(op.Quantity) AS TotalProductsOrdered,
    SUM(CASE WHEN os.OrderStatus = 'Complete' THEN op.Quantity ELSE 0 END) AS TotalProductsProduced
FROM Orders o
LEFT JOIN OrdersList op ON o.OrderID = op.OrderID
LEFT JOIN OrderStatusHistory os ON o.OrderID = os.OrderID
GROUP BY
    CONVERT(DATE, o.RequestDate)
```

Project Management Issues

1. **Time management.** Struggle finding the right time for the meetings between all team members, as everyone works and we had different available times.
2. **Issue regarding the Orders:**
    a. How to record the orders.
    b. How to properly store and handle all the products that each order includes.
    c. How to track the productions process for each product and the whole order.

Questions

Thank you!