Step 1: Company Selection

1. **Divide into groups of 3** : Form a group of three students to work on this project together.
2. **Choose a company** : The chosen company is a **school store**. This type of business caters to students' needs by providing school supplies such as notebooks, pens, textbooks, etc., and could also offer printing and photocopying services for academic documents.
3. **Company name, description, and purpose** :
    - **Company name**: **EduSupplies**
    - **Description** : EduSupplies is a school store located near a major university. It offers a full range of supplies and equipment for students, from basic stationery to academic books and technical drawing tools.
    - **Purpose** : The main purpose of EduSupplies is to provide students and teachers with all the products necessary for their academic success at affordable prices and with efficient service.

Step 2 : Requirements Gathering - Create a List of Requirements

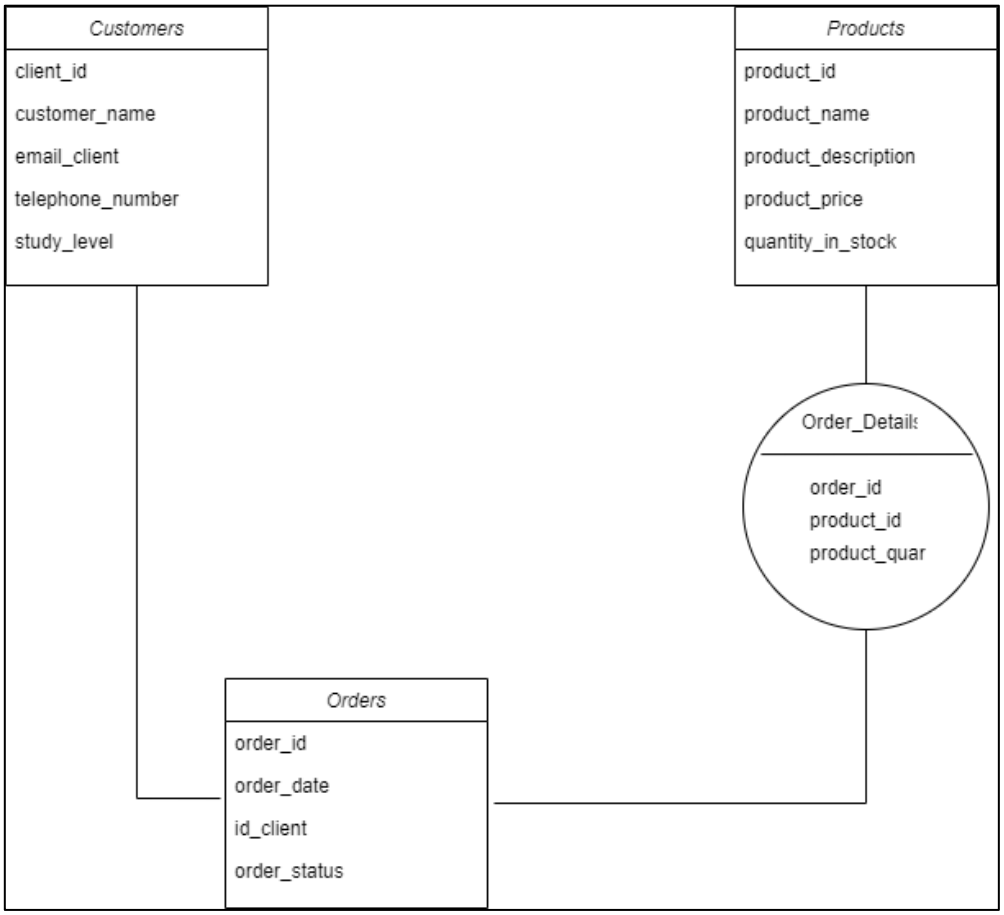
1. **What products or services does your company offer ?**
    - **Products/Services** : School supplies (notebooks, pens, rulers, pencils, pencil cases, calculators), textbooks, snacks, school clothing (uniforms, T-shirts).
2. **What information do you need to store about customers ?**
    - **Customer Information** : Full name, educational level, student ID, email address, phone number, purchase history.
3. **What kinds of transactions will your company handle ?**
    - **Transactions** : Sales (online and in-store payments), refunds, special orders for items not in stock.
4. **What reports or summaries will your company need ? (Optional)**
    - **Reporting Needs** : Daily sales totals, best-selling items, current inventory, list of regular customers, refund summaries.

Step 3 : Data Flow Diagram (DFD) (15 points)

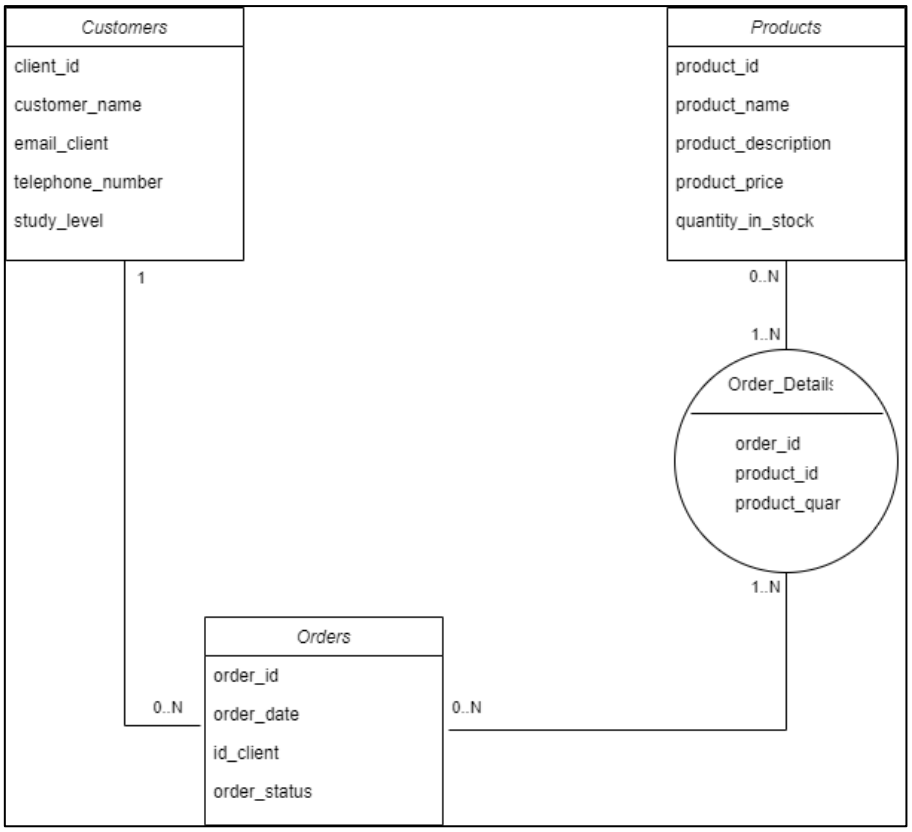
1. **Define external entities (sources/destinations of data)** :
    - **Customers** : Sources of orders and payments.
    - **Suppliers** : Sources for stock replenishment.
    - **Payment management system** : Destination for financial transactions.
2. **Identify processes (actions performed on data)** :
    - **Order process** : Receiving customer orders.
    - **Payment process** : Processing customer payments.
    - **Stock management process** : Updating stock after each order.

- **Refund process** : Handling refund requests.

3. **Determine data stores (storage of data)**:
    - **Customer database**: Contains customer information (name, ID, contact).
    - **Product database**: Contains product list, availability, and prices.
    - **Transaction database**: Stores sales and refund history.
4. **Draw data flows**:



Step 4: Entity-Relationship Diagram (ERD)



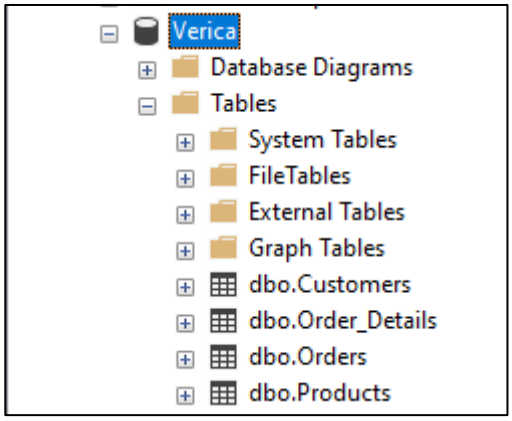
Step 5: Table Creation (SQL)

```sql
-- Customers table
CREATE TABLE Customers (
    client_id INT PRIMARY KEY,
    customer_name VARCHAR(100) NOT NULL,
    email_client VARCHAR(100) UNIQUE NOT NULL,
    telephone_number VARCHAR(15),
    study_level VARCHAR(50)
);

-- Products table
CREATE TABLE Products (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(100) NOT NULL,
    product_description TEXT,
    product_price DECIMAL(10, 2) NOT NULL,
    quantity_in_stock INT NOT NULL
);

-- Orders table
CREATE TABLE Orders (
    order_id INT PRIMARY KEY,
    order_date DATE NOT NULL,
    client_id INT,
    order_status VARCHAR(50),
    FOREIGN KEY (client_id) REFERENCES Customers(client_id)
);

-- Order_Details table
CREATE TABLE Order_Details (
    order_id INT,
    product_id INT,
```

100 %

Messages

Query executed successfully.



Step 6: Data Insertion (SQL)

```sql
-- Insert data into Customers table
INSERT INTO Customers (client_id, customer_name, email_client, telephone_number, study_level)
VALUES (1, 'Alice Dupont', 'alice.dupont@example.com', '0123456789', 'University');

-- Insert data into Products table
INSERT INTO Products (product_id, product_name, product_description, product_price, quantity_in_stock)
VALUES (1, 'Blue Pen', 'Blue ink ballpoint pen', 1.50, 200);

-- Insert data into Orders table
INSERT INTO Orders (order_id, order_date, client_id, order_status)
VALUES (1, '2024-11-04', 1, 'In Progress');

-- Insert data into Order_Details table
INSERT INTO Order_Details (order_id, product_id, product_quantity)
VALUES (1, 1, 2);
```

```
100 %
Messages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)
```

Step 7: Queries and Data Manipulation (SQL)

1. **« SELECT » statements**

```sql
-- Select all customers
SELECT * FROM Customers;

-- Select orders for a specific customer
SELECT * FROM Orders WHERE client_id = 1;
```

100 %

Results | Messages

	client_id	customer_name	email_client	telephone_number	study_level
1	1	Alice Dupont	alice.dupont@example.com	0123456789	University

	order_id	order_date	client_id	order_status
1	1	2024-11-04	1	In Progress

2. **« INSERT », « UPDATE », « DELETE » statements**

```sql
-- Insert a new product
INSERT INTO Products (product_id, product_name, product_description, product_price, quantity_in_stock)
VALUES (2, 'Notebook', '100-page notebook', 2.00, 150);

-- Update the price of a product
UPDATE Products
SET product_price = 1.75
WHERE product_id = 1;

-- Delete an order
DELETE FROM Order_Details
WHERE order_id = 1;


DELETE FROM Orders
WHERE order_id = 1;
```

100 %

Messages

```
(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)
```

3. « JOIN » and aggregate functions

```sql
-- Display order details with customer and product names
SELECT C.customer_name, P.product_name, D.product_quantity, O.order_date
FROM Order_Details D
JOIN Orders O ON D.order_id = O.order_id
JOIN Customers C ON O.client_id = C.client_id
JOIN Products P ON D.product_id = P.product_id;

-- Calculate total sales per product
SELECT P.product_name, SUM(D.product_quantity * P.product_price) AS total_sales
FROM Order_Details D
JOIN Products P ON D.product_id = P.product_id
GROUP BY P.product_name;
```

100 %

Results | Messages

customer_name	product_name	product_quantity	order_date

product_name	total_sales