SQL Code:

```
CREATE TABLE Students (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50),
    age INT,
    course VARCHAR(50),
    marks INT
);
```

This SQL statement creates a Students table with:

- id: Unique identifier (auto-incremented primary key)
- name: Student's name
- age: Integer
- course: Course name
- marks: Integer marks

- Product -- Id, Barcode, Brand, Price, ManuDate, ExpDate
- Customer -- ID, Firstname, Lastname, EmailId
- Course -- ID, CourseName, Instructor

These lines show the structure or attributes for common entities — probably to help with creating similar tables or classes.

- DDD -- Domain Driven Design
- BDD -- Behavior Driven Design
- TDD -- Test Driven Design

`These are software design and testing methodologies.`

`My understanding:`
The SQL tables (`Students`, `Product`, `Customer`, etc.) are structured similarly to classes in OOP.

Just like we store data in tables, in OOP, we store data in objects.

Why do we still use OOP (Object-Oriented Programming)?
We still use OOP because it's practical, intuitive, and excellent for building large, maintainable software systems.

We create objects to bring class blueprints to life — to store data and behavior in a modular way.