

# CS 202: Semester Project - Part 1

Fall 2025

Database Management Systems

## Important Information

**Project Groups:** Groups of 2 students.

**Deadline for Part 1:** November 19, 2025, 11:59 PM

**Deadline for Part 2:** (To be announced later)

**Strict Deadline Policy:** The deadline is absolute. There will be **no extensions, no grace period, and no exceptions.**

Copying or submitting any part of existing online projects will result in an immediate **penalty of -50 points** and will be reported for academic misconduct.

There will be an **in-person demo session** after Part 2 submission (the date and location will be announced later). Each group will be required to present their work in approximately 15 minutes and answer technical questions about their project.

### TAs' Email Addresses:

- Tuğçe Özgirgin: [tugce.ozgirgin@ozu.edu.tr](mailto:tugce.ozgirgin@ozu.edu.tr)
- Azra Oymaağac: [azra.oymaagac@ozu.edu.tr](mailto:azra.oymaagac@ozu.edu.tr)

## Honor Code

As part of our commitment to academic integrity and ethical conduct, all students undertaking the Database Management Systems project are expected to adhere to the following honor code:

1. **Original Work:** All submissions must be the original work of the group. Plagiarism or any form of unauthorized collaboration is strictly prohibited.
2. **Citations and References:** Properly cite and reference any external sources, including books, articles, websites, or any other materials used in the project. Failure to acknowledge sources is considered a violation of academic honesty.
3. **Independent Effort:** Each group is expected to complete the project independently without seeking unauthorized help from other students, online sources, or any external parties unless explicitly permitted by the instructor.
4. **Honesty and Authenticity:** The submission must accurately represent the group's own understanding and effort, reflecting the knowledge and skills gained throughout the course.
5. **Respect for Academic Integrity:** Uphold the principles of academic integrity and abide by the rules and guidelines provided by the instructor and the institution.
6. **Adherence to Course Policies:** Follow all guidelines, deadlines, and instructions specified in the project description and course syllabus. Non-compliance may result in penalties.
7. **Report Violations:** Students are encouraged to report any suspected violations of the honor code to the instructor for appropriate investigation and action.

**Consequences of Violating the Honor Code:** Violations will result in disciplinary action as per the course or institutional policies. Penalties may include, but are not limited to, failing grades on the project, failing the course, or academic probation.

By submitting their project, each student acknowledges their understanding and commitment to upholding this honor code.

# 1. Description

In this project, you are expected to design and develop the database schema for an **E-Commerce Order Management System**. The goal of this assignment is to apply the theoretical and practical database design concepts covered in the course, such as entity-relationship modeling, functional dependency analysis, normalization, and SQL schema creation.

Your design should clearly demonstrate that the system satisfies **Third Normal Form (3NF)** requirements and includes appropriate **constraints, primary and foreign keys, and referential integrity rules**.

## System Overview:

The **E-Commerce Order Management System** should represent the core operations of an online shopping platform. The database must capture information about users, products, orders, payments, and inventory. Your system should include the following conceptual components:

- **User Management:** Maintain user profiles and access roles. At least three distinct user types should exist:
  - **Customer** – browses products, places orders, and makes payments.
  - **Seller** – manages product listings, inventory, and fulfills orders.
  - **Administrator** – oversees users, categories, and system-level configurations.
- **Product Management:** Store information about products, including name, category, price, description, and stock quantity. Each product should have a unique identifier (ProductID).
- **Order Management:** Record orders made by customers. Each order should reference the customer who placed it, the ordered products, order date, total amount, and order status (e.g., pending, shipped, delivered, canceled).
- **Payment Management:** Keep track of payment transactions, including payment ID, payment date, amount, and payment method (e.g., credit card, transfer, wallet). Each order must be linked to its payment record.
- **Inventory Management :** Associate each product with its stock level and ensure updates occur when new stock arrives or when products are sold.

The design report requires the following tasks:

- Finding out the information requirements of the **E-Commerce Order Management System**. Determine the constraints and domains. Determine the entities and relations.
- Identifying the properties of the entities and their domains.
- Determining the identifier of each entity (i.e., find the primary key).
- Drawing the **Entity-Relationship (ER) diagram**.
- Deciding what the base relations are.
- Drawing the **Functional Dependency diagrams**.
- Making sure your relations are in **3NF**.
- Deciding what the referential integrity constraints are (identify foreign keys).
- Deciding which deletion integrity rules to use (**RESTRICT**, **SET NULL**, or **CASCADE**).
- Considering frequent access, come up with a physical database schema.

## DDL and DML File Requirements:

- Create the database on a DBMS (MySQL).
- Specify all **DDL statements** in SQL (covering internal, external, and conceptual levels).
- Specify the **DML statements** needed for the transactions you determined in the design phase.
- Write **SQL insertion, deletion, modification, and selection** statements to demonstrate how the database operates.

**Next Phase:** The implementation and demonstration of this database through **MySQL and Java JDBC** will be conducted in **Part 2**. Further implementation details will be announced later along with the Part 2 instructions.

---

## 2. Design Requirements

You are expected to design the schema for an **E-Commerce Order Management System** that models the core operations of an online marketplace. Your design must capture the entities, relationships, and rules described below. This project focuses on **data modeling, normalization, and integrity**.

### 2.1 Core Entities and Concepts

Your database should represent the key components of an e-commerce system, capturing the entities and relationships necessary to manage users, products, transactions, and post-sale activities. While the high-level concepts below should be considered in your design, you are expected to analyze them carefully and determine an appropriate, normalized schema.

- **Users**, each assigned one of several roles such as customer, seller, or administrator, depending on their responsibilities and access level in the system.
- **Products**, which are offered by sellers and contain various attributes such as product name, description, and pricing details. In addition, availability-related information — including current stock levels, restock status, or inventory updates — should be considered carefully in your design. Each product should also be associated with classification labels such as product category (e.g., Electronics, Clothing, Books). How inventory-related data is managed and tracked may influence how you structure this part of the schema.
- **Orders**, created by customers to request one or more products from a seller.
- **OrderItems**, which associate products with the orders they belong to, along with quantities and price information at the time of purchase.
- **Payments**, representing the financial transactions made toward orders, including method and status.
- **Addresses**, provided by users for shipping and billing purposes.
- **Catalogs**, which sellers use to organize and manage the products they currently offer.
- **Shipments**, which track the delivery process for orders once they are fulfilled by the seller.
- **Reviews**, submitted by customers to share feedback and ratings for the products they have received.

You are expected to critically evaluate whether certain information should be abstracted into distinct entities to ensure a well-structured and normalized design.

### 2.2 Relationships and System Rules

Your design should reflect and enforce the following relationships:

- A user may have multiple addresses.
- Each order must be associated with at least one appropriate address.
- Each product must belong to exactly one category.
- Each seller must have exactly one catalog, and each catalog must belong to a single seller.
- Catalogs contain the products offered by their respective sellers.
- Only administrators should be able to create or modify product categories.
- Customers should be able to place orders containing one or more products.
- A single order may include multiple products, but they must all belong to the same seller. Orders involving products from multiple sellers must be placed separately.
- Orders must reference the customer who placed them and the payment(s) associated with them.
- Shipments must be linked to an order and reflect its delivery progress. Shipment status updates are managed by administrators.
- Customers may submit reviews only for products they have purchased and that have been marked as shipped. Each review should be associated with the relevant order item and customer.
- Your design should support status updates for orders, payments, and shipments as customers and sellers progress through the purchasing and fulfillment process.

You should consider how relationships and foreign keys will enforce these behaviors within your schema.

### 2.3 Normalization Requirement (3NF)

Your database design must follow **Third Normal Form (3NF)**. You are responsible for identifying functional dependencies within your entities and ensuring that your design avoids redundancy and update

anomalies.

You should carefully examine whether certain information (such as category details or repeated product attributes) belongs in separate relations. Your report should include justification for your normalization decisions.

## 2.4 Integrity and Deletion Policies

Your system must maintain logical consistency and referential integrity. Consider the following while designing your schema:

- You should consider how your schema will behave when users, sellers, or products are removed or deactivated.
- Think about which relationships should prevent deletion, which may allow nullification, and which might cascade changes.
- You must document and justify the referential integrity choices you apply in your design.

Your report should clearly state how your chosen policies ensure consistency across the system.

## 2.5 Stock Handling and Status Tracking

Your system should allow accurate handling of inventory and status changes throughout the order lifecycle. Consider how your design can support:

- Updating stock levels when orders are confirmed.
- Preventing overselling when product quantities reach zero.
- Reflecting the progression of order and payment statuses (e.g., payment completion, seller fulfillment).
- Managing shipment status, including confirmation that an order has been shipped and delivered.
- Allowing customers to submit product reviews only after their associated order has been marked as shipped.
- Ensuring that both customers and sellers have visibility into order progress throughout the lifecycle.

You should explain how your schema supports these behaviors without prescribing specific interface or implementation details.

## 3. Submission Guideline For Part 1

1. Please compress all your files and submit a single zip file.
2. The name of your zip file should look like the following: '**Group X Project Part 1.zip**'
3. Your submission should have the following structure:
  - **Group X Project Part 1.zip**
    - Report.pdf
    - DDL.sql (CREATE TABLE statements)
    - DML.sql (INSERT statements)
4. Your database design must comply with **Third Normal Form (3NF)**.
5. Write a report detailing your design decisions.
6. Your report **MUST** include:
  - An **ER diagram** representing your database design.
  - The **functional dependencies** used in your design.
  - A clear explanation of your design decisions and the reasoning behind them.
  - The **names of all group members**.
7. Failing to comply with these guidelines will result in a **penalty of (-5)** for each missing part.
8. Your submission must include a **DDL.sql** file containing all relevant **CREATE TABLE** statements.