Backstory: The Great EU Pizza Scandal 🍕🇪🇺

It all started in Brussels, in what the media now calls **The Great Pizza Scandal of 2027**.

For years, a few massive online pizza-ordering platforms had quietly taken over the market. Almost every pizza in Europe — from a Naples Neapolitan to a Finnish reindeer-topped pie — was ordered through the same two or three apps.

Then came the **investigation**.
EU regulators discovered that these platforms were:

- Charging hidden *“oregano taxes”* to restaurants
- Replacing real mozzarella with *“digitally enhanced cheese-like product”* in photos
- Using AI to **auto-swap your order** to “whatever was cheapest for them that day”
- And worst of all… secretly ranking Brussels-style endive-and-anchovy pizza as “Most Popular” in all countries.

The final straw came when **the platforms accidentally declared pineapple pizza an EU cultural heritage dish**.

The public outrage was immediate. MEPs debated for 14 hours straight.
One impassioned speech ended with the now-famous line:

“We must take back control… slice by slice!” 🍕

The result: **EU Regulation 2028/PI-ZZ-A**.
Centralized pizza ordering was banned.
Every pizzeria must now run its **own independent ordering system** — no shared apps, no shady algorithms, no pineapple unless you *really* want it.

That’s where **you** come in.
Mamma Mia’s Pizza has hired you to build their database-driven ordering system so they can thrive in this brave new post-scandal pizza economy.

Project Description

You will work in two connected layers:

1. Database Layer

- Design an ERD
- Create a relational schema with constraints
- Populate tables with realistic data
- Write SQL queries and transactions to answer the restaurant's needs

2. Programming Layer

- Small console program (or minimal web UI)
- Uses SQL queries to:
    - Display menu
    - Place orders
    - Apply discounts
    - Assign delivery drivers
    - Generate staff reports

Deliverables

At the end of the 6 weeks, you will submit:

1. **Final ERD** (PDF or image)
2. **SQL schema** (CREATE TABLE statements with constraints)
3. **Sample data** (INSERT statements)
4. **SQL queries** (reports, discounts, advanced queries, transactions)
5. **Program source code** (Java, Python, or another approved language)
6. **Video presentation** (5–10 min)

! This project is designed for 2 people. Please go to the People page in Canvas and join a group with another student. !

Video Presentation Requirements

Your **only graded submission** is a recorded presentation (max 10 minutes) in which you:

- Introduce your project

- Explain your database design using your ERD
- Show the database in action
    - Run a few SQL queries directly in your DB client
- Demonstrate your application:
    - Place an order, apply a discount, assign a driver
    - Show at least two staff reports
    - Highlight transactions and constraints
    - Show what happens if invalid data is inserted
- Conclude with what you learned

You may use screen recording + voiceover or appear on camera if you wish.

✅ Project Functional Specification Checklist

Use this checklist to track what your pizza system supports. All **core features** must be implemented. Bonus features are optional, but great for depth and higher marks.

🍕 Core Features (Required)

📋 Menu & Pricing

- Menu displays pizzas, drinks, and desserts from the database
- Pizzas are linked to ingredients (many-to-many)
- Pizza price is **calculated dynamically**:
    - Ingredient costs
        - 40% margin
        - 9% VAT
- No price column is stored directly in the `pizza` table
- Vegetarian/vegan labels are shown based on ingredients

🧑‍💼 Customer & Order Management

- Customers can be added with full info (name, birthdate, address, etc.)
- Orders can be placed with:
    - One or more pizzas (mandatory)
    - Optional drinks and desserts
- Order confirmation includes all items and total price

🎁 Discounts & Loyalty

- Customers get 10% off after 10 pizzas (tracked over time)
- One-time discount codes can be redeemed (only once)
- On their birthday, customers get:
  - 1 free pizza (cheapest)
  - 1 free drink
- Discounts are applied dynamically in SQL or application logic

---

🚚 Delivery Assignment

- Delivery personnel are assigned to postal codes
- Orders are assigned to a delivery person based on customer’s postcode
- A delivery person becomes unavailable for 30 minutes after delivery
- Delivery status is tracked and visible

📊 Reports (Staff Interface)

- Undelivered orders are displayed
- Top 3 pizzas sold in the past month
- Earnings reports filtered by:
  - Gender
  - Age group
  - Postal code

🔄 Database Transactions & Constraints

- Placing an order is wrapped in a **transaction**
- Rollback occurs if any part of the order fails
- Constraints are enforced:
  - Vegetarian pizzas don’t contain meat
  - Valid date of birth
  - Ingredient costs are > 0
  - Discount codes are unique and single-use

✨ Bonus Features (Optional but encouraged)

💡 Database Design & Querying

- Use **views** to encapsulate logic (e.g. menu with prices, veg filters)
- Use **stored functions/procedures** (e.g. for price calculation, discount logic)
- Create and use **indexes** to optimize slow queries
- Use **check constraints** creatively to enforce business rules

---

⏱️ Time & Temporal Logic

- Use timestamps to manage:
  - Delivery timing
  - Order cancellation window (e.g., cancel within 5 minutes)
- Monthly/weekly sales queries use SQL date/time functions

🧪 Testing & Data Generation

- Generate realistic test data using SQL scripts or Python tools (e.g. Faker)
- Include at least 20 sample orders across different timeframes

---

🛡️ Design Thinking & Security (Discussed or Implemented)

- Discussed or implemented access control ideas (e.g. staff vs customer)
- Prevented obvious forms of misuse or data entry mistakes

🎥 Final Video Presentation (All Parts are Required)

Make sure your final video:

- Shows your **ERD** and explains your design
- Demonstrates placing an order and applying a discount
- Shows pizza price calculation working correctly
- Shows at least **2 staff reports** running live
- Shows 1 transaction in action (including rollback if you dare)
- Ends with a reflection on what you learned or what surprised you

---

Week-by-Week Plan

Week 1 – Data Modeling & Project Kickoff

Goals:

- Understand the project context (EU pizza scandal)
- Identify main entities and relationships
- Normalize the data model
- Practice writing ER diagrams

Activities:

- Read project brief + backstory
- Identify functional requirements
- Create ERD with:
  - Pizza, Ingredient, Customer, Order, OrderItem, DeliveryPerson, DiscountCode
- Apply normalization principles
- Discuss how pizza pricing will be calculated dynamically

Deliverables:

- ER diagram
- Written description of business rules (discount logic, delivery constraints)

Week 2 – Schema Design & Database Connection in Code

Goals:

- Convert ERD into a relational schema
- Implement the schema with proper constraints
- Set up code to interact with the database

Activities:

- Write SQL CREATE TABLE statements
- Add primary keys, foreign keys, CHECK constraints, and uniqueness conditions
- Connect your application (Python, Java, etc.) to the database
- Run a simple SELECT query from code

Deliverables:

- SQL DDL script
- Application connects to DB and retrieves menu data
- One sample view: e.g., pizza price view (ingredients → sum + margin + VAT)

Week 3 – Sample Data & Menu Functionality

Goals:

- Populate database with realistic data
- Implement menu display functionality
- Calculate and show pizza prices dynamically

Activities:

- Write INSERT statements:
  - At least 10 pizzas, 10 ingredients, 10 customers
  - 3 delivery persons with assigned postal codes
  - Drinks and desserts
- From your application:
  - Display full menu
  - Show calculated pizza prices
  - Label pizzas as vegetarian/vegan
- Introduce views to encapsulate pricing logic

Deliverables:

- Sample data insert script
- Working menu in application with calculated prices
- Pizza price SQL view

Week 4 – Orders, Discounts & Delivery Assignment

Goals:

- Implement order placement
- Track customers and pizza count
- Apply birthday and loyalty discounts
- Assign delivery personnel

Activities:

- Create orders with pizzas and extras
- Link orders to customer records
- Apply logic for:
  - 10-pizza loyalty discount
  - One-time discount codes
  - Birthday freebies
- Use SQL to assign delivery person based on postal code
- Implement delivery “cooldown” logic (30 mins unavailable)

Deliverables:

- Orders can be placed via application
- Discounts and delivery assignments work correctly
- SQL logic handles delivery person constraints

Week 5 – Reports, Transactions & Constraints

Goals:

- Add reporting features for staff
- Implement transactions for safe order placement
- Enforce strict data integrity

Activities:

- Write complex SQL queries:
  - Top-selling pizzas

    - Undelivered orders
    - Monthly earnings by gender, age, postal code
- Wrap order placement in a database transaction
- Use rollback logic if something fails
- Test constraints:
    - Invalid ingredients in vegetarian pizzas
    - Reused discount codes
    - Negative ingredient prices

Deliverables:

- Staff reports working via SQL + application
- Transactions working correctly
- Constraints verified (e.g. broken rules are caught)

Week 6 – Polish & Final Video Presentation

Goals:

- Finalize application and schema
- Prepare and record the demo
- Reflect on what you learned

Activities:

- Clean up code and SQL
- Prepare rich, varied sample data for the demo
- Record a 5–10 min video presentation showing:
    - ERD explanation
    - Menu with calculated prices
    - Order placement
    - Discount logic
    - Delivery assignment
    - Reporting queries
    - Constraints in action
    - One transaction (with or without rollback)

Deliverables:

- Final application code
- Final SQL schema and data
- Video presentation