This project focused on designing a fully normalized database system for XYZ Company, which handles employees, customers, departments, products, sales, job applications, interviews, and part vendors.

Understanding:
The system required handling multiple complex relationships — recursive (supervisors), overlapping subtypes
(people playing multiple roles), multivalued data (phone numbers), and M:N connections (products and parts,
people and job applications, etc.). Modeling this required a strong EER structure and careful schema design.

Challenges:
- Modeling the generalization hierarchy (Person as a superclass) while ensuring no data duplication
- Resolving many-to-many relationships and tracking historical data like department assignments
- Designing SQL queries that join multiple entities and filter correctly based on business logic

Assumptions Made:
- Each interview has one interviewer and one interviewee
- Email is an implied field in Person for query requirements
- Job selection is based on average grade > 70 and at least 5 passed rounds (grade > 60)
- A supervisor is always an employee and must exist for each non-CEO employee

What I Learned:
- How to construct normalized schemas from EER models
- SQL view creation and complex join/filter logic
- How to maintain relational integrity using keys and constraints
- How to represent historical and recursive data effectively in relational form

This project helped reinforce data modeling principles, SQL implementation, and real-world schema planning in a
multi-entity business context.