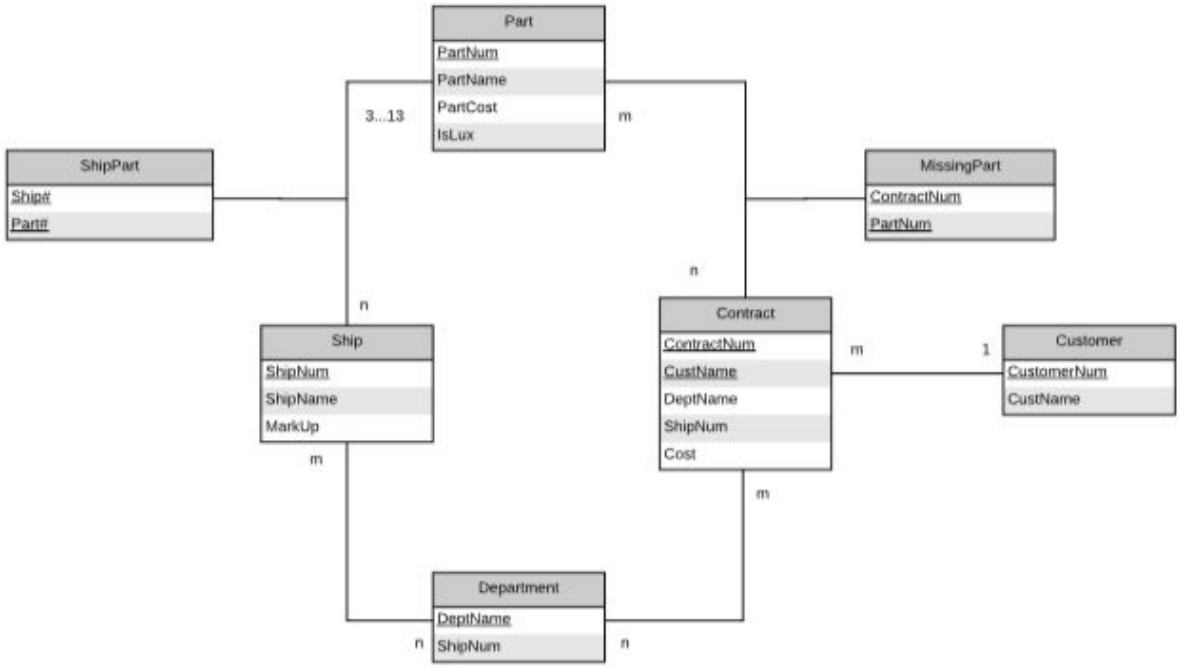
Conceptual Database Design

Four vital entities to the ship company are Part, Ship, Customer and Department. Customers ask departments to build ships, and ships are made of parts. Not all customers want the same ship, so we have the Contract entity to keep track of which customer wants which ship. Similarly, not every ship has the same parts, so we have the ShipPart entity to keep track of which parts can be added to which ship. Finally, not every contract will have the same parts added even if the ships being built are the same, so we need MissingPart to keep track of which contracts require which parts.



Note: Since each ship has three necessary parts and three to ten luxury parts, a ship can have a minimum of three parts (only necessary parts) and a maximum of thirteen parts (necessary parts and all ten luxury parts), hence the 3..13 multiplicity from Ship to Part.

Logical Database Design

The tables resulting from the diagram in the previous section result in the following tables:

Customer	CustNum	CustName

Part	PartNum	PartName	PartCost	IsLux

Department	DeptName	ShipNum

Ship	ShipNum	ShipName	MarkUp

ShipPart	ShipNum	PartNum

Contract	ContractNum	CustNum	DeptName	ShipNum	Cost

MissingPart	ContractNum	PartNum

Normalization Analysis

Below are the functional dependencies in our database:

CustNum → CustName
PartNum → PartName
PartNum → PartCost
PartNum → isLux
DeptName → ShipNum
ShipNum → ShipName
ShipNum → MarkUp
ContractNum → CustNum
ContractNum → DeptName
ContractNum → ShipNum
ContractNum → Cost
ContractNum → PartNum

According to the definition of 3NF, for every non-trivial functional dependency X → A that holds in R, either (a) X is a superkey of R, or (b) A is a prime attribute of R. The relations Customer, Part, Department, Ship, Contract, and MissingPart only contain function dependencies of the form X → A where X is a superkey. This satisfies part (a) of the definition of 3NF. The only exception is the relation ShipPart, which includes a composite key. This still satisfies part (b) of the 3NF definition. Due to the nested nature of the normal forms, we can deduce our schema falls under 2NF and 1NF as well.

Query Description

Query 1: For a given contract, what is the most expensive part? Give its name and the cost. This query gives a customers a rough idea of how expensive a missing part is and it allows them to prioritize the missing part.

Query 2: The user gives a contract number and the amount they intend to pay each month. Display the number of months it will take to pay off a contract. This query shows customers how long they can expect to pay off their ship.