

Customer { CustNo, FirstName, LastName, Email }

Department { DeptName, Model, LaborCost }

Part { PartNo, PartName, Price }

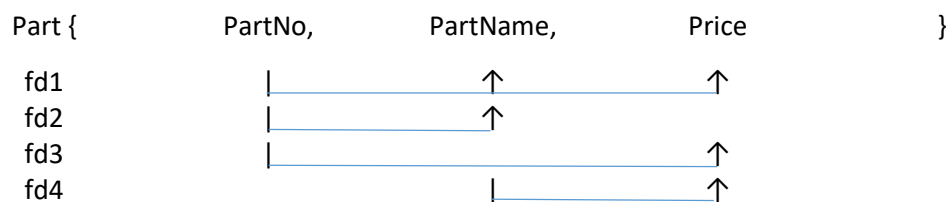
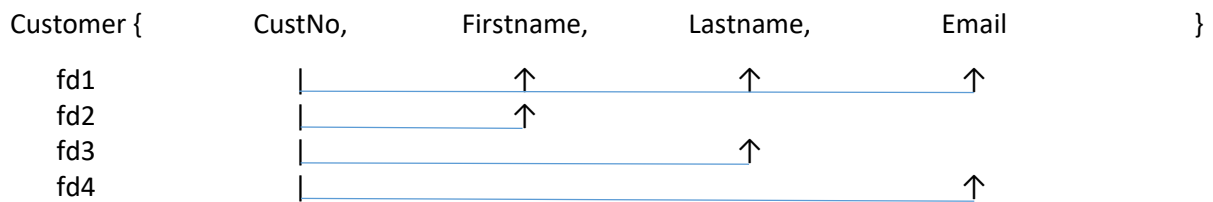
Contract { ContractNo, ContractDate, CustNo }

ContractOrder { OrderNo, ContractNo, DeptName, Status }

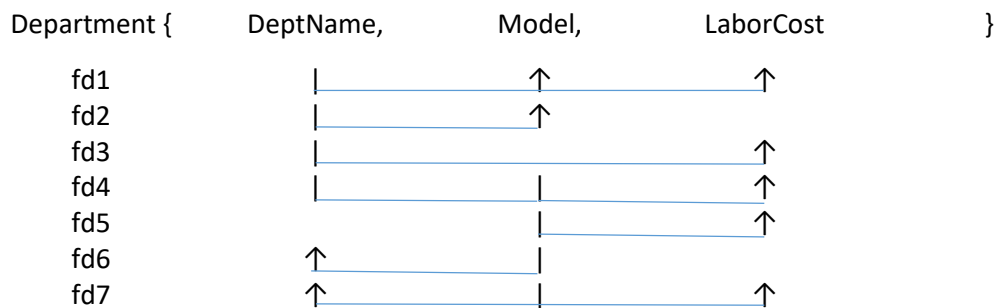
Build { OrderNo, PartNo, Installed }

Customer, Department, and Part are strong entities, therefore their FDs and PKs are trivial.

Customer and Part are in BCNF because CustNo, and PartNo are superkey for the relations.



One department is responsible for building one type of ship, therefore the candidate keys for the Department relation are DeptName and Model, because fd2 and fd4. We use DeptName as primary key for Department relation. This relation is in BCNF because every determinant is a candidate key in the relation.

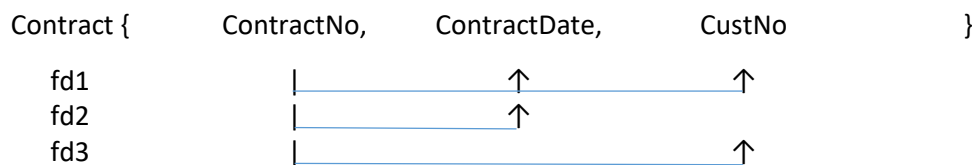


Contract { ContractNo, CustNo, Date, DeptName, Model }

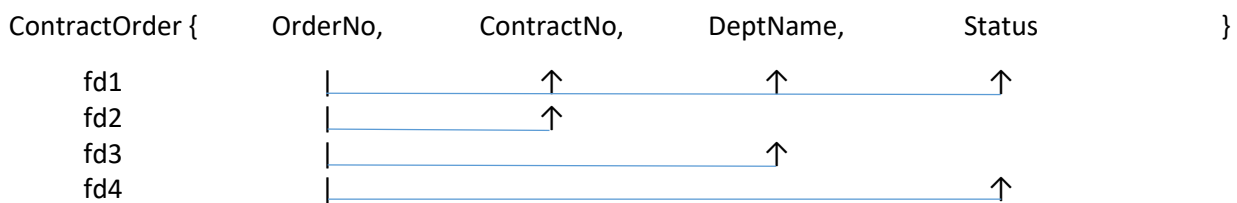
When a customer sign a contract with I^2 , customer may have multiple ship orders in one contract, therefore we use Contract to store contract related information, and ContractOrder to store the order details related with each contract.

The original Contract relation contained details of a customer's order. There is a problem with the original design, there are too many redundancy information stored in the table. A customer can order multiple ships in the same date with different/same departments with different number of features, what described here is multi-valued dependency, therefore we decomposed the Contract relation into Contract and ContractOrder relations. Contract stored information related with the contract, and ContractOrder stored order details related with each contract. This way, the contract related information only stored once. The ContractOrder still contained redundancy problem, because each contract can have n orders, the ContractNo will appears n times in ContractOrder. This is a good tradeoff, because in original design there are $2 * n$ redundancy (Date and CustNo), also provided useful information about the contract detail.

The Contract relation is in BCNF, because ContractNo is a superkey of this relation.



ContractOrder is in BCNF because OrderNo is a superkey of this relation.



Build table store information about a particular order

