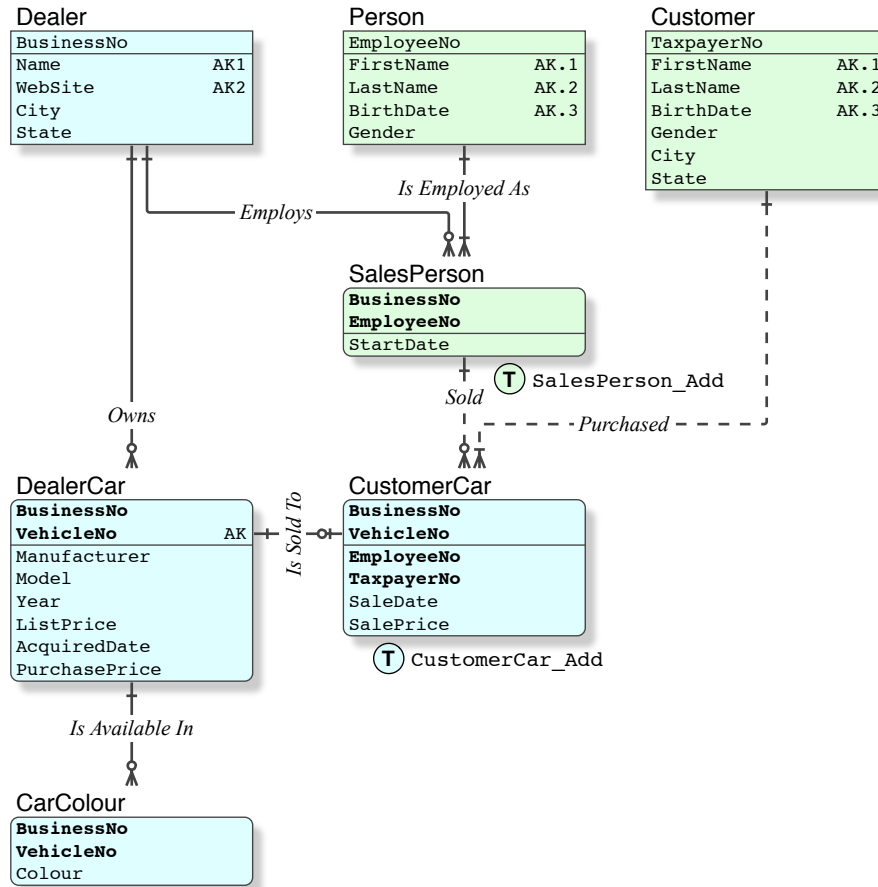
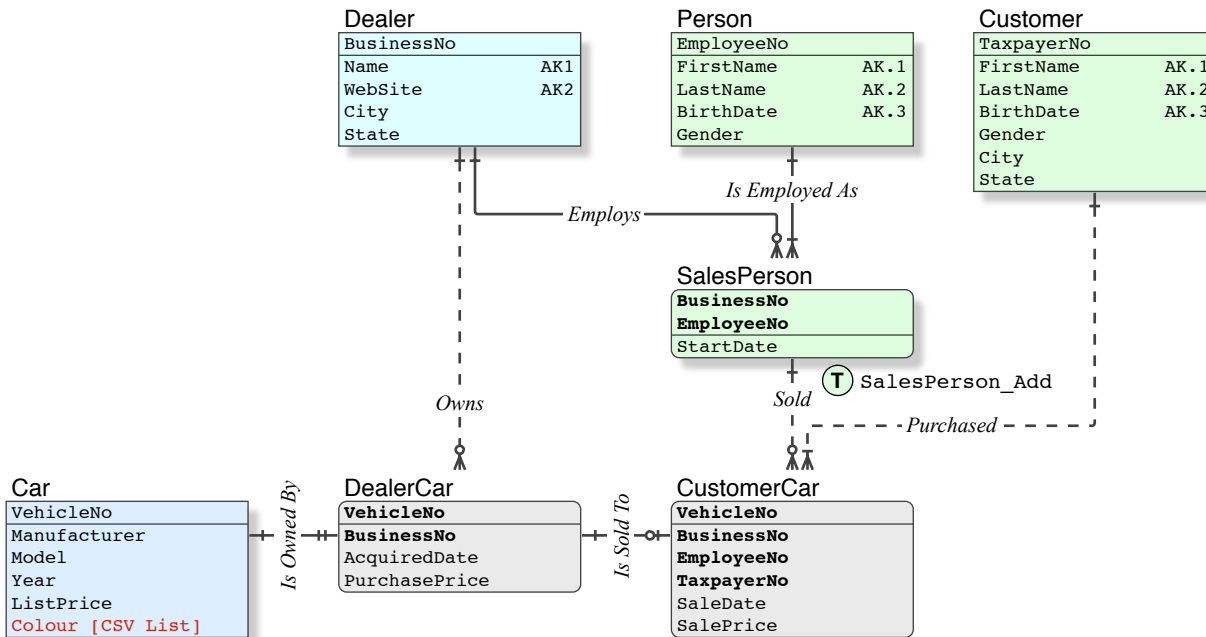


Relational Data Model • Table Attribute

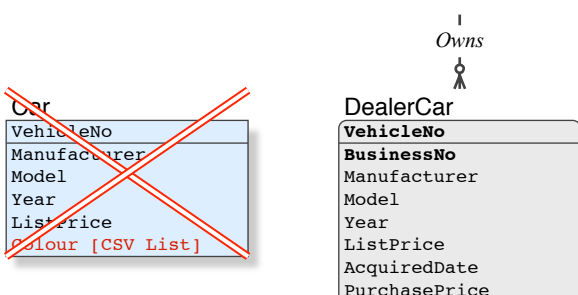


Data Model • Table Attribute • Relational Fail



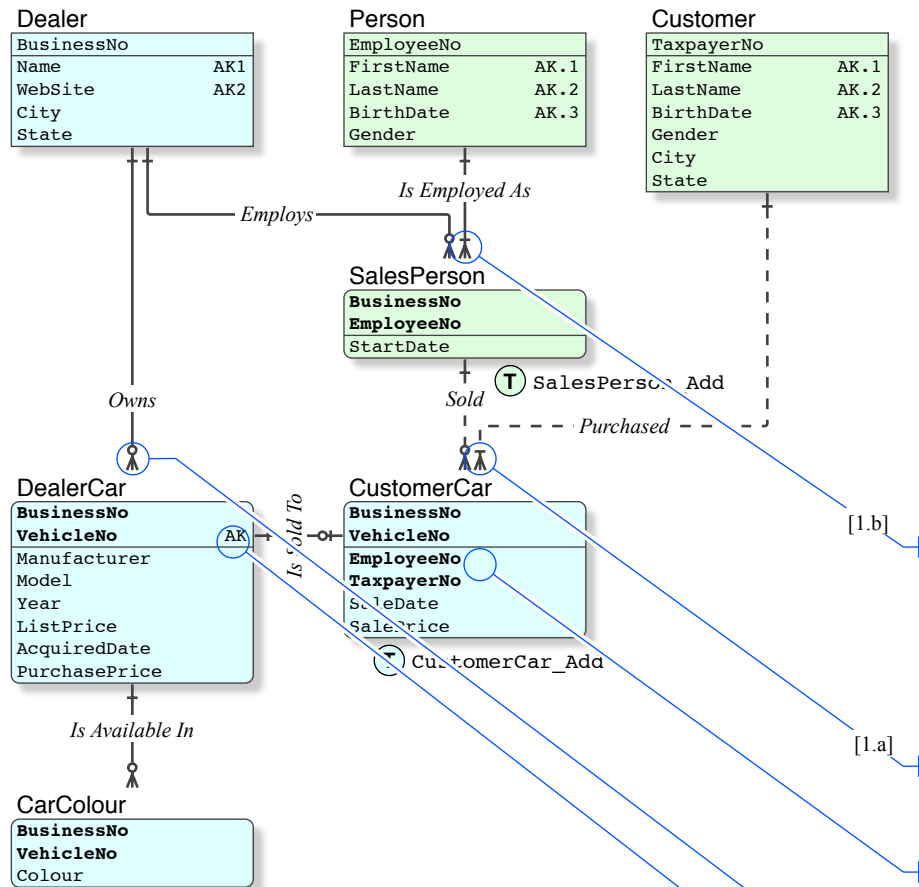
- Slavishly implements the requirement as given
- Fails Normalisation: all attributes that are 1::1 with the PK must be in the same row
- Fails "must be owned by a dealership in order to be in the database"
- Colour Fails: breaks 1NF
- SalesPerson can sell a Car that belongs to any Dealer (not constrained to a Dealer where he is a SalesPerson)
- Customer can purchase a Car from one Dealer & SalesPerson that does not belong to that Dealer

Data Model • Table Attribute • Improved, but still Fails



- Normalisation fixed
- Fixed "must be owned by a dealership in order to be in the database"
- But fails in CustomerCar

### Relational Data Model • Constraint



### Requirement • Rule

**1. Car Dealerships**  
have a unique business id number, dealership name, city, state, and a unique website URL. The business id is a number in the [range from 1000-9999](#).

**2. Salespersons**  
have a first name, last name, [age](#), gender, and a unique employee id number.  
  
The employee number is in the [range 100-999](#).

**3. Customers**  
have a first name, last name, city, state, gender, [age](#), and a unique taxpayer id number. The taxpayer id number is [9 digits](#).

**4. Cars**  
have a make, model, year, suggested price, and a unique vehicle id number.  
**A car can have (possibly multiple) colors that should be modeled as an attribute.**  
Note: the make of a car is the manufacturer (e.g., Ford, Honda, BMW) and the model is the name of the model (e.g.Civic, Accord, CRX).

**5. Salespersons**  
work for dealerships. In order to be in the database, **a salesperson must work for a dealership.**  
However, a dealership may exist in the database without any salespersons. Dealerships have many salespersons who work for them, and a salesperson may work for many different dealerships. The database should record the start date when a salesperson began working for a dealership.

**8. Salespersons sell cars to customers.**  
Even if a salesperson has not sold any cars to customers they should still be stored in the database. However, the database should **only store information about customers who have purchased a car from a salesperson.**  
Information about cars owned by dealerships should be stored in the database regardless of whether they have been sold or not. A salesperson may sell a particular car to only one customer.  
**Similarly, a customer may purchase a particular car from only one salesperson.**  
However, a customer may purchase more than one car from the same salesperson. When a salesperson sells a car to a customer, the sales price and date should be recorded in the database.

**7. Dealerships own cars.**  
**A dealership may own many cars, or they may be completely out of inventory and own zero cars.**  
**A car can only be owned by one dealership and must be owned by a dealership in order to be in the database.**  
The database should record the date that the dealership acquired the car and the price that the dealership paid for the car.

### SQL Level Constraint/Note

- CREATE DATATYPE [2] BusinessNo SMALLINT CHECK ( 1000 >= business\_id <= 9999)
- Storing age and other relative values is an error: the database will require update to all rows every year. Facts such as BirthDate are permanent. Age can be derived from that.
- CREATE DATATYPE [2] EmployeeNo SMALLINT CHECK ( 100 >= business\_id <= 999)
- CREATE DATATYPE [2] TaxpayerNo DECIMAL( 9, 0 ) CHECK ( 100000000 >= business\_id <= 999999999)
- That breaks 1NF:
  - **Each column must be Atomic wrt the platform**
  - No, I will not model an error, the model contains the correct method
- Since the Car is unique (AK: VehicleNo),
  - it can be sold by only 1 Salesperson (BusinessNo)
- Therefore, a Car is not Independent (No Car table), it exists only in the context of a Dealer (BusinessNo, VehicleNo)
- The uniqueness of a Car (VehicleNo) is preserved by the AK
- Since the Car is unique (AK: VehicleNo)
  - it can be owned by only 1 Dealer (BusinessNo)
  - it can be sold to only 1 Customer (TaxpayerNo)

### Note

- 1 Refer to the [Subtype](#) document (§ **1 Implementation: Relationship** only) for a proper understanding of Relationships, and how Cardinality is implemented.
  - All writes to the database should be via Transactions only, thus they have to be written
  - If you have an SQL platform, it has ACID Transactions, which is the simple and correct method implement Cardinality Rules
  - If you have a pretend-sql suite that is MyNONsql, InnoDB supports Transactions (but not true ACID)
  - If you have a pretend-sql suite that is PusgreNONsql, it has no Transactions; no ACID, but Functions are "transactional".
  - a. Eg. there would not be a Customer\_Add Transaction, the CustomerCar\_Add Transaction would INSERT the Customer if he does not exist.
  - b. Eg. there would not be a Person\_Add Transaction, the SalesPerson\_Add Transaction would INSERT the Person if he does not exist.
- 2 For constraints such as Datatype and range, it is best to create a Datatype, named for the Key, and used wherever the Key is. The syntax for the SQL that is required to create a Datatype, is dependent on the particular platform (SQL flavour).
- 3 Alternate Keys are given for Keys that are missing or neglected.