Step one: Create strong entity

```
CREATE TABLE customer (
  custID VARCHAR(255),
  name VARCHAR(255),
  accountBalance INT,
  PRIMARY KEY(custID)
);

CREATE TABLE stuff (
  stuffID VARCHAR(255),
  name VARCHAR(255),
  jobTitlt VARCHAR(255),
  salary INT,
  PRIMARY KEY(stuffID)
);

CREATE TABLE supplier (
  supplierID VARCHAR(255),
  name VARCHAR(255),
  price FLOAT(2),
  PRIMARY KEY(supplierID)
);

CREATE TABLE order (
  orderNumber VARCHAR(255),
  orderquantity INT,
  status VARCHAR(25),
  PRIMARY KEY(orderNumber)
);

CREATE TABLE pricing (
  state VARCHAR(99),
  parameter FLOAT(2),
  PRIMARY KEY(state)
);

CREATE TABLE product (
  productID VARCHAR(255),
  productName VARCHAR(255),
  productInfo VARCHAR(255),
  productSize FLOAT(2),
  url VARCHAR(255),
  PRIMARY KEY(productID)
);

CREATE TABLE category (
  category VARCHAR(255),
  type VARCHAR(255),
  PRIMARY KEY(category, type)
);

CREATE TABLE warehouse (
  warehouseID VARCHAR(255),
  capacity FLOAT(2),
  PRIMARY KEY(warehouseID)
);

CREATE TABLE address (
  addressID VARCHAR(255),
  addressLine VARCHAR(255),
  state VARCHAR(99),
  city VARCHAR(99),
  zip INT,
  PRIMARY KEY(addressID)
);
```

Step two: Create weak entity

```
CREATE TABLE customer (
  custID VARCHAR(255),
  name VARCHAR(255),
  accountBalance INT,
  PRIMARY KEY(custID)
);

CREATE TABLE stuff (
  stuffID VARCHAR(255),
  name VARCHAR(255),
  jobTitlt VARCHAR(255),
  salary INT,
  PRIMARY KEY(stuffID)
);

CREATE TABLE supplier (
  supplierID VARCHAR(255),
  name VARCHAR(255),
  price FLOAT(2),
  PRIMARY KEY(supplierID)
);

CREATE TABLE order (
  orderNumber VARCHAR(255),
  orderquantity INT,
  status VARCHAR(25),
  PRIMARY KEY(orderNumber)
);

CREATE TABLE pricing (
  state VARCHAR(99),
  parameter FLOAT(2),
  PRIMARY KEY(state)
);

CREATE TABLE product (
  productID VARCHAR(255),
  productName VARCHAR(255),
  productInfo VARCHAR(255),
  productSize FLOAT(2),
  url VARCHAR(255),
  PRIMARY KEY(productID)
);
```

```
CREATE TABLE category (
  category VARCHAR(255),
  type VARCHAR(255),
  PRIMARY KEY(category, type)
);

CREATE TABLE warehouse (
  warehouseID VARCHAR(255),
  capacity FLOAT(2),
  PRIMARY KEY(warehouseID)
);

CREATE TABLE address (
  addressID VARCHAR(255),
  addressLine VARCHAR(255),
  state VARCHAR(99),
  city VARCHAR(99),
  zip INT,
  PRIMARY KEY(addressID)
);

CREATE TABLE creditcard (
  cardNumber INT,
  validDate DATE,
  addressID VARCHAR(255),
  custID VARCHAR(255),
  PRIMARY KEY(cardNumber, addressID, custID),
  FOREIGN KEY(addressID) REFERENCES address,
  FOREIGN KEY(custID) REFERENCES customer
)
```

Step three: no multiple attributes

Step four: Create relationships

```
CREATE TABLE customer (
  custID VARCHAR(255),
  name VARCHAR(255),
  accountBalance INT,
  PRIMARY KEY(custID)
);

CREATE TABLE order (
  orderNumber VARCHAR(255),
  state VARCHAR(99),
  orderquantity INT,
  status VARCHAR(25),
  PRIMARY KEY(orderNumber),
  FOREIGN KEY(state) REFERENCES pricing,
  FOREIGN KEY(quantity) REFERENCES cart
);

CREATE TABLE supplier (
  supplierID VARCHAR(255),
  addressID VARCHAR(255),
  name VARCHAR(255),
  price FLOAT(2),
  PRIMARY KEY(supplierID),
  FOREIGN KEY(addressID) REFERENCES address
);

CREATE TABLE product (
  productID VARCHAR(255),
  supplierID VARCHAR(255),
  category VARCHAR(255),
  productName VARCHAR(255),
  productInfo VARCHAR(255),
  additionalInfo VARCHAR(255),
  productSize FLOAT(2),
  url VARCHAR(255),
  PRIMARY KEY(productID),
  FOREIGN KEY(supplierID) REFERENCES supplier,
  FOREIGN KEY(category) REFERENCES category
);

CREATE TABLE pricing (
  state VARCHAR(99),
  parameter FLOAT(2),
  PRIMARY KEY(state)
);

CREATE TABLE stuff (
  stuffID VARCHAR(255),
  addressID VARCHAR(255),
  name VARCHAR(255),
  jobTitlt VARCHAR(255),
  salary INT,
  PRIMARY KEY(stuffID),
  FOREIGN KEY(addressID) REFERENCES address
);

CREATE TABLE warehouse (
  warehouseID VARCHAR(255),
  addressID VARCHAR(255),
  capacity FLOAT(2),
  PRIMARY KEY(warehouseID),
  FOREIGN KEY(addressID) REFERENCES address
);

CREATE TABLE cart (
  custID VARCHAR(255),
  productID VARCHAR(255),
  cartquantity INT,
  orderNumber VARCHAR(255),
  PRIMARY KEY(custID, productID),
  FOREIGN KEY(custID) REFERENCES customer,
  FOREIGN KEY(productID) REFERENCES product,
  FOREIGN KEY(orderNumber) REFERENCES order
);


CREATE TABLE stock (
  warehouseID VARCHAR(255),
  productID VARCHAR(255),
  price FLOAT(2),
  PRIMARY KEY(warehouseID, productID),
  FOREIGN KEY(warehouseID) REFERENCES warehous,
  FOREIGN KEY(productID) REFERENCES product
);


CREATE TABLE delivery (
  custID VARCHAR(255),
  addressID VARCHAR(255),
  PRIMARY KEY(custID, addressID),
  FOREIGN KEY(custID) REFERENCES customer,
  FOREIGN KEY(addressID) REFERENCES address
)

CREATE TABLE category (
  category VARCHAR(255),
  type VARCHAR(255),
  PRIMARY KEY(category, type)
);

CREATE TABLE address (
  addressID VARCHAR(255),
  addressLine VARCHAR(255),
  state VARCHAR(99),
  city VARCHAR(99),
  zip INT,
  PRIMARY KEY(addressID)
);
```