Relations

Companies(id, name, hqAddress, dateOfCreation, country)
FD:
id -> name, hqAddress, dateOfCreation, country
name, hqAddress, dateOfCreation -> id, country

NO DECOMPOSITION NEEDED. Everything on the left side is a superkey.

Products(productId, companyId, name, picture, description, price, quantityInStock, categoryName)
FD:
productId -> categoryName, companyId, name, picture, description, price, quantityInStock
companyId, name -> productId

NO DECOMPOSITION NEEDED. Everything on the left side is a superkey.

Categories(name, createdAt, deletedAt)
FD:
name -> createdAt, deletedAt

NO DECOMPOSITION NEEDED. Everything on the left side is a superkey.

Items(id, quantity, productId)
FD:
id -> quantity, productId

NO DECOMPOSITION NEEDED. Everything on the left side is a superkey.

Orders(id, status, createdAt, deliveryId, itemId, email) (* considering the user is able to provide only one email)
FD:
id -> status, createdAt, deliveryId, productId, email
deliveryId, productId, email -> id, createdAt

(HAVING SOME TROUBLE HERE, SINCE THE STATUS CAN CHANGE WE MIGHT END UP WITH MULTIPLE ROWS FOR THE SAME ORDER WHICH WE CAN'T ALLOW)

If we assume status is static, we can say
NO DECOMPOSITION NEEDED. Everything on the left side is a superkey.

Deliveries(id, deliveredAt, shippedAt, type, addressId)
FD:
id -> deliveredAt, shippedAt, type, addressId
deliveredAt, shippedAt, addressId - > type, id

NO DECOMPOSITION NEEDED. Everything on the left side is a superkey.

Users(email, phoneNumber, dateOfBirth, type, createdAt, defaultAddressId, sex, lastname, firstname, password)
FD:
email -> phoneNumber, dateOfBirth, type, createdAt, defaultAddressId, sex, lastname, firstname, password
phoneNumber, lastname, firstname -> email
MVD:
email ->-> phoneNumber

NO DECOMPOSITION NEEDED. No FD or MVD is violating any BCNF or 4NF conditions.

Addresses(id, userEmail, streetNumber, streetName, city, apartmentNumber, country, zipcode, state, type)
FD:
id -> userEmail, streetNumber, streetName, city, apartmentNumber, country, zipcode, state, type
userEmail, streetNumber, streetName, city, apartmentNumber, country -> zipcode, state, type, id
zipcode, country -> state, city

here the third FD is violating BCNF. So, decomposing the original relation will give

Addresses(id, userEmail, streetNumber, streetName, apartmentNumber, country, zipcode)
Demographics(zipcode, country, state, city)

(A weak entity set and at the "many" side of the many to one relation to the entity set Users. Constraint here is no two users can have the same address.)