The Normalization Process

We can apply the principles just described to formulate the following **normalization process** for normalizing relations:

1. Identify all the candidate keys of the relation.
2. Identify all the functional dependencies in the relation.
3. Examine the determinants of the functional dependencies. If any determinant is not a candidate key, the relation is not well formed. In this case:
   a. Place the columns of the functional dependency in a new relation of their own.
   b. Make the determinant of the functional dependency the primary key of the new relation.
   c. Leave a copy of the determinant as a foreign key in the original relation.
   d. Create a referential integrity constraint between the original relation and the new relation.
4. Repeat step 3 as many times as necessary until every determinant of every relation is a candidate key.

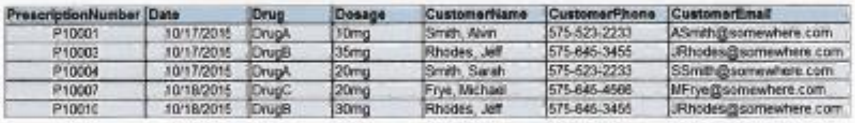
To understand this process, consider the following relation:

PRESCRIPTION (PrescriptionNumber, Date, Drug, Dosage, CustomerName, CustomerPhone, CustomerEmail)

Sample data for the PRESCRIPTION relation are shown in Figure 2-17.

FIGURE 2-17

Sample PERSCIPTION Relation and Data



PrescriptionNumber	Date	Drug	Dosage	CustomerName	CustomerPhone	CustomerEmail
P10001	10/17/2015	DrugA	10mg	Smith, Alvin	575-523-2233	ASmith@somewhere.com
P10003	10/17/2015	DrugB	35mg	Rhodes, Jeff	575-645-3455	JRhodes@somewhere.com
P10004	10/17/2015	DrugA	20mg	Smith, Sarah	575-523-2233	SSmith@somewhere.com
P10007	10/18/2015	DrugC	20mg	Frye, Michael	575-645-4566	MFrye@somewhere.com
P10010	10/18/2015	DrugB	30mg	Rhodes, Jeff	575-645-3455	JRhodes@somewhere.com

Step 1 of the Normalization Process According to the normalization process, we first identify all candidate keys. PrescriptionNumber clearly determines Date, Drug, and Dosage. If we assume that a prescription is for only one person, then it also determines CustomerName, CustomerPhone, and CustomerEmail. By law, prescriptions must be for only one person, so PrescriptionNumber is a candidate key.

Does this relation have other candidate keys? Date, Drug, and Dosage do not determine PrescriptionNumber because many prescriptions can be written on a given date, many prescriptions can be written for a given drug, and many prescriptions can be written for a given dosage.

What about customer columns? If a customer had only one prescription, then we could say that some identifying customer column—for example, CustomerEmail—would determine the prescription data. However, people can have more than one prescription, so this assumption is invalid.

Given this analysis, the only candidate key of PRESCRIPTION is PrescriptionNumber.

Step 2 of the Normalization Process In step 2 of the normalization process, we now identify all functional dependencies. PrescriptionNumber determines all the other attributes, as just described. If a drug had only one dosage, then we could state that:

Drug → Dosage

But this is not true because some drugs have several dosages. Therefore, Drug is not a determinant. Furthermore, Dosage is not a determinant because the same dosage can be given for many different drugs.

However, examining the customer columns, we do find a functional dependency:

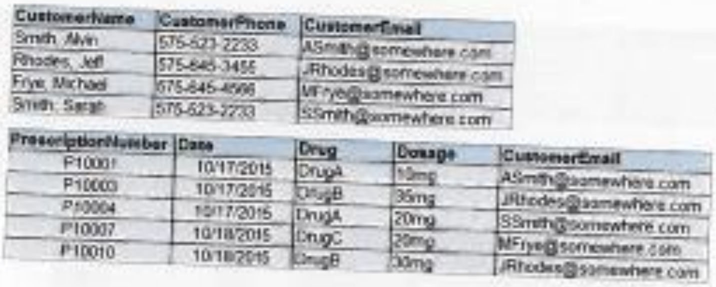
CustomerEmail → (CustomerName, CustomerPhone)

To know whether functional dependency is true for a particular application, we need to look beyond the sample data in Figure 2-17 and ask the users. For example, it is possible that some customers share the same email address, and it is also possible that some customers do not have email. For now, we can assume that the users say that CustomerEmail is a determinant of the customer attributes.

Step 3 of the Normalization Process In step 3 of the normalization process, we ask whether there is a determinant that is *not* a candidate key. In this example, CustomerEmail is a determinant and not a candidate key. Therefore, PRESCRIPTION has normalization problems and is not well formed. According to step 3, we split the functional dependency into a relation of its own:

CUSTOMER (CustomerName, CustomerPhone, <u>CustomerEmail</u>)

We make the determinant of the functional dependency, CustomerEmail, the primary key of the new relation.



CustomerName	CustomerPhone	CustomerEmail
Smith, Alvin	575-523-2233	ASmith@somewhere.com
Rhodes, Jeff	575-645-3455	JRhodes@somewhere.com
Frye, Michael	575-645-4566	MFrye@somewhere.com
Smith, Sarah	575-523-2233	SSmith@somewhere.com

PrescriptionNumber	Date	Drug	Dosage	CustomerEmail
P10001	10/17/2015	DrugA	10mg	ASmith@somewhere.com
P10003	10/17/2015	DrugB	35mg	JRhodes@somewhere.com
P10004	10/17/2015	DrugA	20mg	SSmith@somewhere.com
P10007	10/18/2015	DrugC	20mg	MFrye@somewhere.com
P10010	10/18/2015	DrugB	30mg	JRhodes@somewhere.com

FIGURE 2-18

Normalized Prescription Customer Relations and Data

We leave a copy of CustomerEmail in the original relation as a foreign key. Thus, PRESCRIPTION is now:

PRESCRIPTION (PrescriptionNumber, Date, Drug, Dosage, *CustomerEmail*)

Finally, we create the referential integrity constraint:

CustomerEmail in PRESCRIPTION must exist in CustomerEmail in CUSTOMER

At this point, if we move through the three steps, we find that neither of these relations has a determinant that is not a candidate key, and we can say that the two relations are now normalized. Figure 2-18 shows the result for the sample data.