

What are the functional dependencies?
 What are the candidate keys?

Type	Class	Size	Color	Shape	Length
1	A	Large	Blue	Round	110
2	Br	Small	Blue	Square	100
3	R	Large	Blue	Square	90
2	A	Large	Red	Square	110
4	Br	Small	Red	Triangle	90
3	Br	Small	Green	Square	90
8	R	Large	Blue	Round	110
2	Br	Small	Green	Round	110

Type	Class	Size	Color	Shape	Length
1	A	Large	Blue	Round	110
2	Br	Small	Blue	Square	100
3	R	Large	Blue	Square	80
2	A	Small	Red	Square	110
4	Br	Small	Red	Triangle	90
3	Br	Large	Green	Square	90
8	R	Large	Blue	Round	80
2	Br	Small	Green	Round	70

Functional dependencies: Type → Size, Length → Class

Primary Keys (a unique item, or unique combination of items for each row):
 No single column keys.
 Composite Primary Keys include:
 Type&Color, Type&Length, Shape&Length

Normal forms: another example, courtesy of Phlonx.com
 Our task is to design database tables in 3NF for orders

Assume:
 -Only one name or address per customer

-Only one entry for each item on any given order (e.g., can't have an order with items for 5 Red Freens and 23 Red Freens

The Spreadsheet Solution

Invoice #	Date	Cust. #	Cust. Name	Add. City	State	Zip	Item #	Desc.	Qty	Price	It. #	Desc.	Qty	Price
125	9/13/2002	56	Foo, Inc.	23 Main St.	Thorpeburg	TX	563	56" Blue Freen	4	3.50	851	Spline End (Xtra Large)	32	8.00
126	9/14/2002	2	Freens R	11600 Penn	Washington	DC	563	56" Blue Fr	500	3.5	852	3" Red Freen	750	12

Questions, questions:

How many red freens were ordered in 2002?

What was the total number of 3" red or blue freens sold in California?

What were the names of customers who bought items between July 1 and 5, 2007?

First Normal Forms

Get rid of repeating columns within a row:

Invoice No	Date	Cust No	Cust Name	Cust Add	Cust City	C State	Item ID	Item Desc	Qty	Price	Item ID	Item Desc	Qty	Price
125	9/13/2002	56	Foo, Inc.	23 Main St	Thorpeburg TX	TX	563	56" Blue Fr	4	3.5	851	Spline End	32	0.25
126	9/14/2002	2	Freens R Us	1800 Penn Washington DC	DC	DC	563	56" Blue Fr	500	3.5	652	3" Red Free	750	12

In this case, the item description, quantity, price, and description

Invoice No	Date	Cust No	Cust Name	Cust Address	Cust City	Cust State	Item ID	Item Description	Item Qty	Item Price
125	9/13/2002	56	Foo, Inc.	23 Main St, Thorpeburg	Thorpeburg TX	TX	563	56" Blue Fr	4	\$ 3.5
125	9/13/2002	56	Foo, Inc.	23 Main St, Thorpeburg	Thorpeburg TX	TX	851	Spline End	32	\$ 0.25
125	9/13/2002	56	Foo, Inc.	23 Main St, Thorpeburg	Thorpeburg TX	TX	652	3" Red Free	5	\$ 12.00
126	9/14/2002	2	Freens R Us	1800 Pennsylvania Avenue Washington	DC	DC	563	56" Blue Fr	500	\$ 3.5
126	9/14/2002	2	Freens R Us	1800 Pennsylvania Avenue Washington	DC	DC	652	3" Red Free	750	\$ 12.00

We copy the first columns, and create a new row, for each set of repeating items

For Second Normal Form, we need a primary key – a column or set of columns that allows us to uniquely identify each row

order_id	order_date	customer_id	customer_name	customer_address	customer_city	customer_state	item_id	item_description	item_qty	item_price
125	9/13/2002	56	Foo, Inc.	23 Main St, Thi Thorpeburg	TX	TX	563	56" Blue Freen	4	
125	9/13/2002	56	Foo, Inc.	23 Main St, Thi Thorpeburg	TX	TX	851	Spline End (xtrs	32	
125	9/13/2002	56	Foo, Inc.	23 Main St, Thi Thorpeburg	TX	TX	652	3" Red Freen	5	
126	9/14/2002	2	Freens R Us	1800 Pennsylvka Washington	DC	DC	563	56" Blue Freen	500	
126	9/14/2002	2	Freens R Us	1800 Pennsylvka Washington	DC	DC	652	3" Red Freen	750	

Here, no one column will work, but the combination of order_id (also known as invoice_id) and item_id is a possibility

The structure of the orders table is related to the primary key:

There are those items that change with order_id,

And those items that change with item_id,

But each order/item combination is unique, so they may be use together as a primary key (a concatenated key)

orders
order_id (PK)
order_date
customer_id
customer_name
customer_address
customer_city
customer_state
item_id (PK)
item_description
item_qty
item_price

Second Normal Form:

Every non-key item is F.D. only on the key, 1) either directly, or 2) through a transitive F.D.

How do we test for this?

Test for partial dependence.

Is any column dependent on only part of the primary key?

Is customer_id enough to know the customer name or do we also need to know order_id?

Is item_id enough to know item description, or do we need to know customer_id?

orders
order_id (PK)
order_date
customer_id
customer_name
customer_address
customer_city
customer_state
item_id (PK)
item_description
item_qty
item_price

Second Normal Form:

Every non-key item is F.D. only on the key, 1) either directly, or 2) through a transitive F.D.

How do we test for this?
Test for partial dependence.

Is any column dependent on only part of the primary key?

Is customer_id enough to know the customer name or do we also need to know order_id?

customer_id is enough!

Is item_id enough to know item description, or do we need to know customer_id?

item_id is enough! – not in 2NF

orders

order_id (PK)
order_date
customer_id
customer_name
customer_address
customer_city
customer_state
item_id (PK)
item_description
item_qty
item_price
item_total_price
order_total_price

Table Not in Second Normal Form

How do we get there? We break the table up by

Step1) Projecting over the primary key plus variables dependent only on the primary key,
(remember, projection is just a fancy word for subsetting the columns)

What columns are dependent only on the primary key?

Invoice No.	Date	Cust. No.	Cust. Name	Cust. Address	Cust. City	Cust. State	Item ID	Item Descr	Item Qty.	Item Pri
125	9/13/2002	56	Foo, Inc.	23 Main St., Thorpeburg	Thorpeburg TX		563	56" Blue Fr	4	\$ 3.5
125	9/13/2002	56	Foo, Inc.	23 Main St., Thorpeburg	Thorpeburg TX		851	Spline End I	32	\$ 0.3
125	9/13/2002	56	Foo, Inc.	23 Main St., Thorpeburg	Thorpeburg TX		652	3" Red Free	5	\$ 12.0
126	9/14/2002	2	Freans R Us	1600 Pennsylvania Avenue	Washington DC		563	56" Blue Fr	500	\$ 3.5
126	9/14/2002	2	Freans R Us	1600 Pennsylvania Avenue	Washington DC		652	3" Red Free	750	\$ 12.0

Item quantity

Our first table in 2NF (there are more)

order_id	item_id	item_qty
125	563	4
125	851	32
125	652	5
126	563	500
126	652	750

Step 2) Projecting the table over each functionally dependent group

Invoice No.	Date	Cust. No.	Cust. Name	Cust. Address	Cust. City	Cust. State	Item ID	Item Descr	Item Qty.	Item Pri
125	9/13/2002	56	Foo, Inc.	23 Main St., Thorpeburg	Thorpeburg TX		563	56" Blue Fr	4	\$ 3.5
125	9/13/2002	56	Foo, Inc.	23 Main St., Thorpeburg	Thorpeburg TX		851	Spline End I	32	\$ 0.3
125	9/13/2002	56	Foo, Inc.	23 Main St., Thorpeburg	Thorpeburg TX		652	3" Red Free	5	\$ 12.0
126	9/14/2002	2	Freans R Us	1600 Pennsylvania Avenue	Washington DC		563	56" Blue Fr	500	\$ 3.5
126	9/14/2002	2	Freans R Us	1600 Pennsylvania Avenue	Washington DC		652	3" Red Free	750	\$ 12.0

Here, our functional dependencies are

order_id -> order_date, customer_id,
customer_name,
customer_address, customer_city,
customer_state

item_id -> item_description, item_price

Two more tables, in 2NF

order_id	item_id	item_qty
125	563	4
125	851	32
125	652	5
126	563	500
126	652	750

item_id	item_description	item_price
563	56" Blue Freen	\$3.50
851	Spline End (Xtra Large)	\$0.25
652	3" Red Freen	\$12.00

order_id	order_date	customer_id	customer_name	customer_addr	customer_city	customer_state
125	9/13/2002	56	Foo, Inc.	23 Main St., Thi	Thorpleburg	TX
126	9/14/2002	2	Freens R Us	1600 Pennsylv	Washington	DC

Third Normal Form – remove the transitive functional dependencies

No transitive functional dependencies in these two tables.

order_id	item_id	item_qty
125	563	4
125	851	32
125	652	5
126	563	500
126	652	750

item_id	item_description	item_price
563	56" Blue Freen	\$3.50
851	Spline End (Xtra Large)	\$0.25
652	3" Red Freen	\$12.00

But there are here.

order_id	order_date	customer_id	customer_name	customer_addr	customer_city	customer_state
125	9/13/2002	56	Foo, Inc.	23 Main St., Thi	Thorpleburg	TX
126	9/14/2002	2	Freens R Us	1600 Pennsylv	Washington	DC

order_id -> customer_id, but customer_id -> customer_name, etc.

To create a table in 3NF we need to remove this transitional functional dependency, while maintaining the connections among information.

order_id	order_date	customer_id	customer_name	customer_addr	customer_city	customer_state
125	9/13/2002	56	Foo, Inc.	23 Main St., Thi	Thorpleburg	TX
126	9/14/2002	2	Freens R Us	1600 Pennsylv	Washington	DC

Split the table into:

The functionally dependent items,

customer_id	customer_name	customer_addr	customer_city	customer_state
56	Foo, Inc.	23 Main St., Thi	Thorpleburg	TX
2	Freens R Us	1600 Pennsylv	Washington	DC

The other items

order_id	customer_id	order_date
125	56	9/13/2002
126	2	9/14/2002
0	0	

In summary, four tables from one, but we have

- Reduced storage
- Isolated data, greatly facilitating maintenance, updates
- Identified key relationships
- Structured our data for flexible combinations in queries

order_id	customer_id	order_date
125	56	9/13/2002
126	2	9/14/2002
0	0	

order_id	item_id	item_qty
125	563	4
125	851	32
125	652	5
126	563	500
126	652	750

customer_id	customer_name	customer_addr	customer_city	customer_state
56	Foo, Inc.	23 Main St., Thi	Thorpleburg	TX
2	Freens R Us	1600 Pennsylv	Washington	DC

item_id	item_description	item_price
563	56" Blue Freen	\$3.50
851	Spline End (Xtra Large)	\$0.25
652	3" Red Freen	\$12.00

Steps

- To 1NF – reorganize to remove repeat columns
- Identify a primary key
- Identify the items in the 2NF table that are only partially dependent on the primary key
- Identify the functional dependencies in the 2NF table
- Split the 2NF table – a table for the primary key and fully dependent variables, and a table for each of the clusters of functional dependencies.
- To make 3NF, split any 2NF tables with transitive functional dependencies into separate tables – but remember to keep variables to link these tables