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

<table>
<thead>
<tr>
<th>Type</th>
<th>Class</th>
<th>Size</th>
<th>Color</th>
<th>Shape</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>Large</td>
<td>Blue</td>
<td>Round</td>
<td>110</td>
</tr>
<tr>
<td>2</td>
<td>Br</td>
<td>Small</td>
<td>Blue</td>
<td>Square</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>R</td>
<td>Large</td>
<td>Blue</td>
<td>Square</td>
<td>90</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>Large</td>
<td>Red</td>
<td>Square</td>
<td>110</td>
</tr>
<tr>
<td>4</td>
<td>Br</td>
<td>Small</td>
<td>Red</td>
<td>Triangle</td>
<td>90</td>
</tr>
<tr>
<td>3</td>
<td>Br</td>
<td>Small</td>
<td>Green</td>
<td>Square</td>
<td>90</td>
</tr>
<tr>
<td>8</td>
<td>R</td>
<td>Large</td>
<td>Blue</td>
<td>Round</td>
<td>110</td>
</tr>
<tr>
<td>2</td>
<td>Br</td>
<td>Small</td>
<td>Green</td>
<td>Round</td>
<td>110</td>
</tr>
</tbody>
</table>

Functional dependencies: Type → Size, Length → Class

Primary Keys (a unique item, or unique combination of items for each row):
No single columns keys.
Complete 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)

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Price</th>
<th>Description</th>
<th>Unit Price</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1.94</td>
<td>2nd Red Freen</td>
<td></td>
<td>7.76</td>
</tr>
<tr>
<td>52</td>
<td>1.14</td>
<td>3rd Blue Freen</td>
<td></td>
<td>58.64</td>
</tr>
<tr>
<td>5</td>
<td>1.39</td>
<td>5th Blue Freen</td>
<td></td>
<td>6.95</td>
</tr>
<tr>
<td>2</td>
<td>1.39</td>
<td>3rd Red Freen</td>
<td></td>
<td>2.78</td>
</tr>
<tr>
<td>700</td>
<td>0.06</td>
<td>1st Red Freen</td>
<td></td>
<td>42.00</td>
</tr>
<tr>
<td>100</td>
<td>0.06</td>
<td>2nd Red Freen</td>
<td></td>
<td>6.00</td>
</tr>
</tbody>
</table>

TOTAL DUE: 84.20

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:

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

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

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 used together as a primary key (a concatenated key)

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?
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

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?

Item quantity

Our first table in 2NF (there are more)

Step 2) Projecting the table over each functionally dependent group

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

Third Normal Form – remove the transitive functional dependencies

No transitive functional dependencies in these two tables.

But there are here.

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

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
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