The table below shows a sample booking for a guest at the XYZ chain of hotels:

<table>
<thead>
<tr>
<th>Hotel No</th>
<th>Guest No</th>
<th>HotelName</th>
<th>City</th>
<th>Hotel Zip</th>
<th>GuestName</th>
<th>City</th>
<th>Guest Zip</th>
<th>Date From</th>
<th>Date To</th>
<th>Room No</th>
<th>Room Type</th>
<th>Room Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>232</td>
<td>Hilton</td>
<td>San Diego</td>
<td>83835</td>
<td>Brad</td>
<td>Knoxville</td>
<td>37996</td>
<td>2012-11-28</td>
<td>2012-12-02</td>
<td>635</td>
<td>King</td>
<td>89.99</td>
</tr>
</tbody>
</table>

You may make the following assumptions about the data:

- A hotel number uniquely identifies a hotel's name and zip code
- A zip code uniquely identifies a city for both hotels and guests (not true in the real world, but true in our fantasy world)
- A guest number uniquely identifies a guest's name and zip code
- A room number and a hotel number uniquely determine a room type and price
- A guest may not have overlapping reservations.
- A room may not be double booked.

Answer the following questions:

- Give an example of the following types of anomalies:
  - insert
  - update
  - delete
- What are the functional dependencies for this relation?
- What are the candidate keys for this relation?
- Show how you would convert this relation to 2nd normal form, and show which functional dependencies you would use to create each new relation.
- Show how you would convert the relations from 2nd to 3rd normal form, and show which functional dependencies you would use to create each new relation.
- What is the name for the type of functional dependency used to convert a relation to 2nd normal form?
- What is the name for the type of functional dependency used to convert a relation to 3rd normal form?