TUTORIAL 5: Normalisation (SOLUTION)

Before they had their system properly designed, GCUTours got their 'IT Guy' to put together a database. After all, how hard could it be to put a few tables together?

IT Guy didn't really know about database design, so he made a bit of a mess of it. In this tutorial you will look at some of his work. You'll think about what he did wrong and about how to fix it.

Question 1

Here's IT Guy's first attempt at a table to store data about holiday **packages**, shown with some sample data:

Packages(packageID, location, name, description, price, departureDates)

packageID	location	name	description	price	departureDates
1	USA	Western Advent	A typical tour is c	£1,499.00	1/3/2008,5/6/2008,2/9/2008
2	Asia	Roof of the Worl	New this year is	£1,599.00	01/03/2008,05/06/2008
3	Europe	Alpine Action	There is adventu	£899.00	1/3/2008,2/9/2008,2/10/2008,3/11/2008

1. **Write an SQL query which will find all holidays with a departure on 5th June 2008.**

   SELECT name, location FROM Packages
   WHERE departureDates LIKE '%5/%6/2008%'

2. **Try writing a query which will list all departure dates and locations – can you do it?**

3. **Summarise the problems that this table design can cause?**

   Problem is a repeating group of data within a field, which causes the following problems:
   - need to extract dates from text string
   - cannot store dates as correct data type
   - need to type dates in exactly the right format

IT Guy tried again:

Packages(packageID, location, name, description, price, departureDate1, departureDate2, departureDate3, departureDate4)

packageID	location	name	description	price	departureD:	departureD:	departureD:	departureD:
1	USA	Western Advent	A typical tour is c	£1,499.00	01/03/2008	05/06/2008	02/09/2008	
2	Asia	Roof of the Worl	New this year is	£1,599.00	01/03/2008	05/06/2008		
3	Europe	Alpine Action	There is adventu	£899.00	01/03/2008	02/09/2008	02/10/2008	03/11/2008

1. **Write an SQL query which will find all holidays with a departure on 5th June 2008**

   SELECT name, location FROM Packages3
   WHERE departureDate1 = '05/06/2008'
   OR departureDate2= '05/06/2008'
   OR departureDate3= '05/06/2008'
   OR departureDate4= '05/06/2008'

2. **In what ways is this better? Is it worse in any ways?**

   Can use correct date type, one item of data per field
   multiple logically related fields (a *repeating group*), required data could be in any of them, what if we need more than 4?

3. **Write descriptions, in the TableName(fieldnames…) notation, of the tables required to get this into 1NF.**

   Packages(packageID, location, name, description, price)
   DepartureDates(departureDate, *packageID(FK)*)

Question 2

IT Guy figured that he needed a table to store **bookings**, so he came up with this:

Bookings(departureDate, name, username, firstname, lastname, numberOfPersons)

departureD	name	username	firstname	lastname	persons
05/02/2008	Colorado Wint	ferdy	Ferdinand	Magellan	3
01/03/2008	Western Adve	mpolo	Marco	Polo	2
01/03/2008	Western Adve	vdagama	Vasco	daGama	4
05/06/2008	Borneo Explor	mpolo	Marco	Polo	1
05/06/2008	Western Adve	ferdy	Ferdinand	Magellan	5

1. **List the functional dependencies in this table.**

   departureDate, name, username → firstname, lastname, persons
   usernname → firstname, lastname

2. **Why is this table not in 2NF?**

   dependency on **username** – part of primary key

3. **Write descriptions, in the TableName(fieldnames…) notation, of the tables required to get this into 2NF.**

   Bookings(departureDate,name,*username(FK)*,persons)
   Users(username,firstname,lastname)

Question 3

IT Guy read something on a website about using ID fields as primary keys, and reckoned that all the problems with the Bookings table would be solved by adding a **bookingID** field to the table.

Bookings(bookingID,departureDate, name, username, firstname, lastname, persons)

bookingID	departureD:	name	username	firstname	lastname	persons
1	05/02/2008	Colorado Wint	ferdy	Ferdinand	Magellan	3
2	01/03/2008	Western Adve	mpolo	Marco	Polo	2
3	01/03/2008	Western Adve	vdagama	Vasco	daGama	4
4	05/06/2008	Borneo Explor	mpolo	Marco	Polo	1
5	05/06/2008	Western Adve	ferdy	Ferdinand	Magellan	5

1. **List the functional dependencies in this table.**

   bookingID → departureDate, name, username, firstname, lastname, persons
   usernname → firstname, lastname

2. **Is this table in 2NF? How do you know?**

   yes it is - no dependencies on part of the primary key (in fact, there couldn't be any, as the PK is a single field)

3. **Why is this table not in 3NF?**

   still have dependency on **username** – now a non-key field (transitive dependency)

4. **Write descriptions, in the TableName(fieldnames…) notation, of the tables required to get this into 3NF.**

   Bookings(bookingID,departureDate,name,*username(FK)*,persons)
   Users(username,firstname,lastname)

Question 4

Here's another of IT Guy's attempts at fixing the table in Question 1. He really doesn't like adding more tables!

Packages(packageID, location, name, description, price, departureDate)

packageID	location	name	description	price	departureDate
1	USA	Western Advent	A typical tour is c	£1,499.00	01/03/2008
1	USA	Western Advent	A typical tour is c	£1,499.00	05/06/2008
1	USA	Western Advent	A typical tour is c	£1,499.00	02/09/2008
2	Asia	Roof of the Worl	New this year is	£1,599.00	01/03/2008
2	Asia	Roof of the Worl	New this year is	£1,599.00	05/06/2008
3	Europe	Alpine Action	There is adventu	£899.00	01/03/2008
3	Europe	Alpine Action	There is adventu	£899.00	02/09/2008
3	Europe	Alpine Action	There is adventu	£899.00	02/10/2008
3	Europe	Alpine Action	There is adventu	£899.00	03/11/2008

1. **What is the primary key of this table?**

   packageID, departureDate

2. **Does this table fix any of the problems identified in Question 1? Does it create any new problems?**

   It fixes most of the problems with querying. However, there is now a lot of repeated data within the table.

3. **This table is still not in 1NF. Why?**

   There are still multiple values of the departuredate field for each set of values of all the other fields, so there is still a *repeating group* (departuredate)

   Fix is the same as in Question 1:

   Packages(packageID, location, name, description, price)
   DepartureDates(departureDate, *packageID(FK)*)

   Check for 2NF would also discover problem with this table and lead to the same resulting pair of tables – think about it!