11. Build your Own Data Models

11.0 Welcome

You are invited to follow developments on our Web Site :-

- [http://www.databaseanswers.org/index.htm](http://www.databaseanswers.org/index.htm)

You can also join our Database Answers Community

- [http://databaseanswers.ning.com/](http://databaseanswers.ning.com/)

Barry Williams
Principal Consultant
Database Answers Ltd.

[barryw@databaseanswers.org](mailto:barryw@databaseanswers.org)

11.1 Introduction

This Chapter provides a Tutorial on how to build your own Data Models that meet a complex set of requirements.

11.1.1 What is this ?

It shows how to progress from simple Models to complex Models in a logical and structured manner.

It starts by analysing some existing Data Models and extending them to meet your own specific requirements.

In this Scenario, we plan these three activities :-

- Opening a Pool Hall
• Opening a Funeral Parlour
• Making a Movie.

11.1.2 Why is it important?
It is important because it is often necessary to understand some existing Data Models and build on them to include additional functionality.

11.1.3 What will I Learn?
You will learn how to build on an existing foundation to create a new set of Data Models that will operate in an enterprise environment.

The Best Practice Approach is defined in these Steps :-

• Establish the User Requirements
• Agree a ‘Statement of Objectives’ with the users, and from this identify the ‘Things of Interest’, the Top-Level Model and the Subject Areas.
• Look for a starting-point.
  o There is nothing new under the sun, and this is certainly true in the world of Data Modelling.
  o Everything is just a variation of something that has been done before.
• From the starting-point, identify the Models that will define your starting-point.
• Create an Information Catalogue to record these Models.
• Define the Naming Standards for Entities and Attributes.
• Define the Scope of your new Model from the User Requirements.
  o In other words, the functional areas and dominant Entities.
• Identify the additional data that you will need which is not in the Models you have chosen to start with.
• Add the data and be sure to follow the appropriate standards because you want your new Model to look good and to pass any QA test.
• Design a normalised Data Model
• Apply the QA Test and produce the Scorecard.
• Discuss the Model and the Scorecard with all the interested parties.
• This will normally include your fellow Modellers, Data Management professionals, Analysts, Developers and Managers.

• Produce Mapping Specifications and plan for migration of the required data.
11.2 Opening a Pool Hall

Where do we start with our Data Model?

Our starting-point is our Library of hundreds of Data Models on the Database Answers Web Site :-

- [http://www.databaseanswers.org/data_models/index.htm](http://www.databaseanswers.org/data_models/index.htm)

We find this Model on Pool League Statistics which might be useful :-

We need a Reservations facility and we have a wide choice:

1. **Reservations** :-
   1. Apartment Rentals
   2. Airline Reservations
   3. Car Hire
   4. Cinema Bookings
   5. Coach Trips
   6. Doctors Practice
   7. Driving School
   8. Event Reservations
   9. Flights for Children's Charities
   10. Hairdressers Appointments
   11. Health Centers
   12. Holiday Homes
   13. Hotel Reservations
   14. Limousine Services
   15. Online Scheduling
   16. Railway Reservations
   17. Restaurant Bookings
   18. Sports Centers
   19. Ticketmaster
   20. Vehicle Rental

Our situation is that we have a fixed number of facilities, ie pool tables, which are normally all available during our opening hours.

Each of our Pool Tables can accommodate up to four players at the same time.

We need to record reservations in our Database and allow for cancellations and changes.

We have quite a number of available Databases to choose from, including these:

1. Apartment Rentals
2. Driving School
3. Hairdressers Appointments
4. Railway Reservations
5. Sports Centers
1. Apartment Rentals

Apartment Rentals Data Model
Barry Williams
DatabaseAnswers.org
2nd. May 2004

Apartment_Buildings
- building_id*
- building_short_name
- building_full_name
- building_description
- building_address
- building_manager
- building_phone
- other_building_details

Ref_Apartment_Facilities
- facility_code*
- facility_description
  eg Broadband, Cable TV.

Ref_Apartment_TYPES
- apt_type_code*
- apt_type_description
  eg Studio, Duplex

Apartments
- apt_id*
- building_id*
- apt_type_code
- apt_number
- bathroom_count
- bedroom_count
- room_count
- other_apartment_details

Apartment_Facilities
- apt_id*
- facility_code*

Ref_Gender
- gender_code*
- gender_description
  Values are M, F or U (unknown)

View_Unit_Status
- apt_id
- status_date*
- available_yn
- apt_booking_id

Apartment_Bookings
- apt_booking_id*
- apt_id
- guest_id*
- booking_status_code*
- booking_status_description
  eg Confirmed or Provisional
- booking_start_date
- booking_end_date
- other_booking_details

Guests
- guest_id*
- gender_code
- guest_first_name
- guest_last_name
- date_of_birth
- other_guest_details
- Driving Schools

This has Lessons which look like what we need, but it also has Customer and Payment details that we don’t need.
- Hairdressers Appointments
- Railway Reservations

This looks too complicated and includes that we don’t need, such as Stations !!!
- **Sports Centers**

  - [http://www.databaseanswers.org/data_models/sports_centers/index.htm](http://www.databaseanswers.org/data_models/sports_centers/index.htm)

  We notice that this Model has code to check on the availability of facilities by the hour.

```sql
The SQL to find available Facilities would look something like this:
SELECT DISTINCT facility_id
FROM bookings
WHERE facility_id NOT IN
  (SELECT facility_id
  FROM bookings
  WHERE booking_start_datetime > required_start
  AND booking_start_datetime < required_end)
```
Step 1. Start with the Customer

First, we start by copying the Sports Centers Model.

We take Cardholders Entity from this Model of Customers and Credit Cards and rename it Customers. We do this simply because it contains a representative number of attributes:

- [http://www.databaseanswers.org/data_models/customers_and_credit_cards/customers_and_credit_cards_with_attributes.htm](http://www.databaseanswers.org/data_models/customers_and_credit_cards/customers_and_credit_cards_with_attributes.htm)

So our Data Model consists of just this Customer Entity:

```
<table>
<thead>
<tr>
<th>Customers</th>
</tr>
</thead>
<tbody>
<tr>
<td>customer_id</td>
</tr>
<tr>
<td>first_name</td>
</tr>
<tr>
<td>last_name</td>
</tr>
<tr>
<td>gender_mfu</td>
</tr>
<tr>
<td>address</td>
</tr>
<tr>
<td>cel_mobile_phone</td>
</tr>
<tr>
<td>other_details</td>
</tr>
</tbody>
</table>
```
Step 2. Add Tables and Bookings

This is done in a style consistent with the Sports Centers Model.

Full payment is made in advance.
Step 3. Add Customers and Payment Methods

Then we add Payment Methods from the Customers and Payments Model:

http://www.databaseanswers.org/data_models/customers_and_payments_e_govt/customers_and_payments_subject_area.htm
Step 4. Add Regular Bookings

This Step adds Repeat Bookings.

Booking Frequencies table is at the top because logically that is where it belongs. However, it gives it an importance that it does not deserve.
Step 5. Add Inheritance

Generic Customers contain common Attributes from Personal and Commercial Customers.

In this design, we store Contact details in First_Name and Last_Name and Gender.

Personal Customers in the US have an SSN (Social Security Number) and in the UK, they have a National Insurance (NI) Number.

In this design, we store Contact details in First_Name and Last_Name and Gender.

Commercial Customers

- Customer_ID
- UK_VAT_Number
- Other_Details
Step 6. Let’s have a Party

This Step introduces Inheritance and the concept of Parties, which are a ‘Super-Type’.

Personal and Commercial Customers are ‘Sub-Types’, along with Staff and Suppliers.
Step 7. Reference Data
This shows all of the Reference Data that we have referred to so far in our Pool Hall Management Data Model.

It is good practice to put Ref_ at the beginning of every Entity or Table.

This makes it very easy to identify the Reference Data.

Some data might be considered Reference Data under certain circumstances.

For example, a Calendar is always Reference Data because it is predictable and never changes. Products, on the other hand, could change regularly and should not be considered Reference Data. For example, Products in finance or banking could change on a weekly basis and are therefore not Reference Data.
Step 8. The Complete Model

This diagram shows all the Tables, all the Keys and Attributes, including Reference Data.

It is suitable for discussion with Developers, Data Analysts and other Data Modellers who want to see all the details of every Table.
Step 9. The Complete Model - showing Key Fields only

This shows all the Tables, with just the Keys.

This is much easier to understand at a glance.

This is suitable for discussion with developers, Data Analysts and other Data Modellers.
Step 11. The Complete Model - showing Entity names only

This shows the minimum possible to explain the Data Model.

This is the easiest to understand at a glance.

This is suitable for discussion with business users, management and other stakeholders who are not interested in the details but want to understand the scope of the Model.
Step 11. The Complete Model - showing Entity names only without Reference Data Tables

This is even easier and is usually acceptable to senior stakeholders.

They will understand the concept of Reference Data but do not need to see it shown explicitly.

One benefit of this approach is that it makes it easy to define the Business Rules, which Users can then understand, agree to and sign-off.

For example, Business Rules would say :-

- Customers can make Regular Bookings, but does not have to.
- Regular Booking must always be associated with a Customer.
Step 11. Top-Level and Subject Area Models

The Subject Areas include:

- Bookings
- Customers
- Payments
- Data Warehouse

13.2.2.1 Top-Level Model
The Top-Level Model goes here.

13.2.2.2 Subject Area Models

13.2.2.3 Bookings
The Bookings Model goes here.

13.2.2.4 Customers
The Customers Model goes here.

13.2.2.5 Payments
The Payments Model goes here.

13.2.2.6 Data Warehouse
The Data Warehouse Model goes here.
11.3 Opening a Funeral Home

Here is another finished product:

- [http://www.databaseanswers.org/data_models/funeral_homes/index.htm](http://www.databaseanswers.org/data_models/funeral_homes/index.htm)

**Step 1. Identify the ‘Things of Interest’**

These include:

- Clients
- Funerals
- Services
- Others to be determined (always useful)

How are they related?

- Clients request Funerals with optional Services.
11.4 Making a Movie

11.4.1 A Simple Model

For this Model, we have a simple finished Model to start with :-

- [http://www.databaseanswers.org/data_models/movie_making/making_a_movie4_beginners.htm](http://www.databaseanswers.org/data_models/movie_making/making_a_movie4_beginners.htm)

This Approach has a fixed number of Stages from Development, through Pre-Production, production to Post-Production and Distribution.

It is easy to work with and efficient which offset the fact that it is a fixed approach.
11.4.2 A Complex Model

Here’s a complex version :-

- [http://www.databaseanswers.org/data_models/movie_making/index.htm](http://www.databaseanswers.org/data_models/movie_making/index.htm)

This shows a different approach to modelling the Stages in making a Movie.

It is more flexible than the fixed Approach because the table called Movie_Stage_Definitions can handle any approach to movie making.

However it is not so easy to work with.

To get around this problem we can use SQL Views to show a fixed set of Stages for any particular Movie.
11.5 What have we learned?

In this Chapter, we have looked at the Steps involved in designing a Data Model with the intention of producing a single integrated Framework with a consistent approach to Entities.

We have learned how to take existing Data Models, then combine and extend them to meet our specific requirements.

This will be very helpful for you if you consider using Data Modelling in your professional life because that situation is very common.

You will therefore emerge looking like a star ;-0)