Data Modeling by Example – a Tutorial

Elephants, Crocodiles and Data Warehouses

Database Answers

Barry Williams
1. Management Summary

1.1 A trip to Malaysia

In this Paper we use a trip to Malaysia to discuss an approach to the implementation of a Reference Data Architecture and the design of a Data Warehouse.

A Canonical Data Model (CDM) is central to this and we discuss the benefits of using Design Patterns based on a CDM.

During the trip, my wife and I stayed in three Hotels, hired a car and visited a number of Tourist Attractions, including an Elephant Sanctuary, a Crocodile Farm and an Underwater World in Langkawi Island in Malaysia.

After we returned to England I found myself thinking that the trip would provide a good opportunity to develop an interesting and ‘User-Friendly’ Tutorial on Data Warehouses.

The design of the Data Models reflects the scope and the fact that the overall aim is to provide data for Business Intelligence.

We also try to keep in mind that a well-designed Data Model should be good to look at and it should be possible to tell a story based on the Model.

1.2 The Approach

The Approach is to follow these Steps for each Event
Data Modeling by Example – a Tutorial

Elephants, Crocodiles and Data Warehouses

Step 1 – Use the Event-driven Canonical Data Model from this page in our Web Site:

Step 2 - Define a Message Format for each Event

Step 3 - Design a Data Warehouse and update it after we analyse each Event.

These Steps will be followed for these Events that occurred during our trip to Langkawi Island in Malaysia:

- Event 1 - Check-in to a Hotel
- Event 2 - Hire a Car
- Event 3 - Go Shopping
- Event 4 - Visit Tourist Attractions
- Event 5 - Check-out of a Hotel

For each Event, we discuss the Operational Data Source (ODS) in the form of the Message for the Event, along with the Data Warehouse and the Dimensional Model.

Our starting-point is defined by the Templates in Section 2.

A short Slide Show has been created to give you an overview in less than one minute:

1.3 Overview from ODS to BI

This Section presents Data Architecture to support a Data Warehouse and BI Layer.

A typical kind of display requested by users is a Pie-chart. This example shows the expenses that we incurred during our trip to Malaysia — Car Hire, Hotel Accommodation, Shopping and Tourist Attractions. The source of the data is Tickets and Sales Receipts generated in Stores. This data is stored in a Data Warehouse to provide a Single View of the Truth for the Pie-Charts in the BI Layer.
2. The Theory
This discussion of the Theory is based on a number of Generic Data Architectures and Templates.

2.1 A Reference Data Architecture
There are many different ways of defining a Data Architecture.

This diagram shows the four major Stages in delivering the ‘Single View of the Truth’ :-

![Data Architecture Diagram]

- Data Sources
- Integrated Data Layer (Clean-up and Transformation)
- Data Warehouse
- Data Marts (for Reports, BI, KPI's etc.)
2.2 A Cloud-based Data Architecture for the Future

This Section describes an Architecture that can be used as a target ‘To-Be’ for the future.

This diagram shows more clearly how Web Services would be used with an ESB providing an Abstraction Layer.

Data Virtualisation is a concept in Implementation that can be defined as ‘the process of providing users with a Business View of the data in an organisation’, which conceals the technical details of stored data, such as location, storage structure, APIs and storage technology.
2.3 Reference Data Architecture

The Integrated Data Platform is a specific example of a more general Data Virtualization Layer.
2.4 Generic Cloud Architecture

This Architecture shows how we will develop the first phase of a Cloud-based Utility.

BI Ready, SQL and Web Services will be used to provide a Integrated Data Platform.

- BI Layer
- Dimensional Models
- 3NF Data Warehouse
- Integrated Data Platform
- Staging Area
- Operational Data Store Layer
  - ODS 1 – Shipments
  - ODS 2 – Service Delivery
  - ODS 3 - Oracle OBPM
  - Etc.
2.5 Template for a Canonical Data Model

The Canonical Data Model is used as a Template for a Design Pattern for an ERD for a Business Event.

This Model appears on this page on our Database Answers Web Site:

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

It provides a ‘stripped-down’ Event-oriented Model that applies to a wide range of business and everyday situations.

We use it as a standard to translate data into a common format suitable for loading into a Data Warehouse.

We have used ERWin for this Data Model.

This allows us to show Many-to-Many Relationships in a very concise and economical style.

When we come to use the CDM we will expand these into One-to-Many Relationships.
2.6 Mapping to the CDM

The purpose of this activity is to identify the correspondence between the Generic entities in the CDM with the specific entities for a particular Event.

This example shows how the Entities in the Data Model for our Hotel Check-In Event map on to our Design Pattern based on our Canonical Data Model (CDM).

We are very happy to see that it does because it helps to validate the CDM.

<table>
<thead>
<tr>
<th>CDM</th>
<th>EVENT : Hotel Check-In</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers</td>
<td>Guests</td>
<td></td>
</tr>
<tr>
<td>Documents</td>
<td>Room Card / Key</td>
<td>This the nearest we get to a Document</td>
</tr>
<tr>
<td>Events</td>
<td>Hotel Check-In</td>
<td></td>
</tr>
<tr>
<td>Locations</td>
<td>Hotel Address</td>
<td></td>
</tr>
<tr>
<td>Organisations</td>
<td>Hotel Chains</td>
<td></td>
</tr>
<tr>
<td>Organisations</td>
<td>Hotels</td>
<td></td>
</tr>
<tr>
<td>Organisations</td>
<td>Staff in the Hotel</td>
<td>Staff can be involved and represent the Organisation</td>
</tr>
<tr>
<td>Products or Services</td>
<td>Room Reservation</td>
<td></td>
</tr>
<tr>
<td>Third Parties</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

2.7 Template Message Format

This is our Generic Template that is used in the derivation of all the specific Message Formats – for Car Hire, Shopping and so on.

<table>
<thead>
<tr>
<th>Generic</th>
<th>Supplier</th>
<th>Date &amp; Time</th>
<th>Customer</th>
<th>Products or Services</th>
<th>Unit Price</th>
<th>From Date</th>
<th>To Date</th>
<th>Total Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific, eg Hotel Check-in</td>
<td>Eg Hotel</td>
<td>Date</td>
<td>Customer, Guest</td>
<td>Room, Meals, etc.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.8 Extract, Transform and Load (ETL)

At this point, we are interested in two aspects of ETL:-

1. Specification of Rules for Validation, Clean-up and Transformation

2. Definition of each item which is in common use and where a clear unique understanding is important.

Therefore, we create a table like this:-

<table>
<thead>
<tr>
<th>Business Term</th>
<th>Validation</th>
<th>Validation</th>
<th>Clean-up</th>
<th>Transform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date &amp; Time</td>
<td>Cannot be in the future</td>
<td>Cannot be before date to be determined, eg 1950</td>
<td>Mark as bad data</td>
<td>Review with SMEs</td>
</tr>
<tr>
<td>Unit Price</td>
<td>Cannot be negative</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>From Date</td>
<td>Less than or equal to “To Date”</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>To Date</td>
<td>Must be before date to be determined, eg 2020</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Price</td>
<td>Cannot be negative</td>
<td></td>
<td>Mark as bad data</td>
<td>Review with SMEs</td>
</tr>
</tbody>
</table>

A wide variety of software is available to meet ETL requirements.

A desirable feature is the ability to build up a library of ETL functions that can be enhanced as a result of learning in the course of a Data Quality Project.

This Library will then provide the foundation for the repetitive processing which is performed continuously until the data reaches the desired quality.

This will be determined by the business users and SMEs.

Although, as data management professionals, we might expect or want to have 100% good data, this is not always strictly necessary.

For example, in UK Local Government, it is common to find that addresses in Council Tax are of a uniform high quality, and that Dates of Birth are good in Social Services, because these two data items are critical to the respective functions.
2.9 Template Data Warehouse

In this Paper we use ERWin Data Models for the Data Warehouses. We shows only the Entity names because it helps to understand the Model.

The Attribute are shown in the Dimensional Models.
During our analysis of the Events we did not find a place for the ‘Third_Parties’ Entity that is shown in our CDM in Section 2.1

Therefore, we do not include in the future discussions in this Paper.
2.10 Template Dimensional Model

This is a Generic Template Dimensional Model which corresponds to our CDM shown above.

This will be an excellent starting-point for the analysis of the required Dimensional Models for the range of Events which make up a very important part of this document.
Data Modeling by Example – a Tutorial
Elephants, Crocodiles and Data Warehouses

2.11 Data Warehouse Bus Architecture
Now, we show the complete Data Warehouse Bus Architecture. It helps us to think through how the Data Architecture will operate at run-time.

It provides a common Framework for establishing consistent standards for data and structures.

The top row shows the major Building Blocks in the migration of data from the Operational Data Stores to the Semantic Layer that defines the BI User View.

```
ODS  Staging Area  3NF Data Warehouse  Dimensional Models  Semantic Layer  BI User View

Data Warehouse Bus

Single View of the Truth

Master Data
Conformed Dimensions
Conformed Facts

eg Calendar – Date
eg Profitability

eg Location
eg Performance
```
2.12 Master Data Management (MDM)

2.12.1 Introduction
The MDM Component provides a Single View of the truth for Customers and Products.

The principles behind the Single View of both the Customers and Products and Customers is the same:

- Establish uniqueness using Business Rules
- Map from alternatives to the single record.
- If possible, define one Data Source as the ‘Golden Record’ or Master Record.
  - For example, where Salesforce is in place, it is often provides the Master Customer record.

In this case, the ‘GetCustomer’ Service will try to match between the new characteristics, such as Customer name and address, and existing Customers.
2.12.2 Customers

A Customer Master Index (CMI) maintains a link between the single master Customer record and the Source Data Customers.

This Data Model shows how the approach above will be implemented in our Reference Data Architecture.

2.12.2.1 The Problem

For this Event, we must plan for loading Customer data into the Data Warehouse.

This requires that we establish a ‘Single View of the Customer’.

In the UK, the name ‘Joe Bloggs’ is used whereas in the States it would be John Doe’.

Joe, of course, is an abbreviation of Joseph.

On official documents, the name would be spelled Joseph, whereas in everyday conversation, it would normally be Joe.

Therefore we have to allow for the possibility that a ‘Joe Bloggs’ might be the same person as ‘Joseph Bloggs’

In the States, we might find the names of John Doe, Johnny Doe and J.Doe and we would have to match them if they are the same person.
2.12.2 The Solution
Resolution of this Problem and the ambiguity requires us to define a set of Business Rules that can be run whenever we load a Customer who might be ambiguous.

Therefore, our solution to this problem of establishing a Single View of Customer Joe or Joseph is to have a ‘Rules Engine’ where we can define and execute a Rule like ‘Joe is equivalent to Joseph’.

The recommended practice to implement the Customer Master Index is to use Web Services for Get, Update and Put facilities.

2.12.3 Products
The requirements here are similar to the Master Customer Index.

We need to match similar products from different Suppliers should be considered the same Product for BI and Performance Reports. For example, Red Sweaters will have different Product Codes from different Suppliers but we will want to include them in the same Product Category that we define for our analysis purposes.

In this way, we establish a Single View of Products and Product Categories in the Data Warehouse.

This will be achieved by a combination of automated Business Rules and manual intervention by members of staff.
3. The Practice

This Section discusses the Events that occurred on our holiday in Malaysia.

3.0 Arriving in Malaysia

We flew from London to Kuala Lumpur (KL).

In the centre of this striking photo are the Petronas Towers in Kuala Lumpur, the capital of Malaysia, and commonly referred to simply as KL. At the time when they were built, they were the highest building in the world, and I believe they are still number two or three, with 80 floors each, and a footbridge on the 40th floor.

I walked across and it is not something I would want to do very often.

In passing, let me say that Petronas is a clever word because it is a combination of Petrol and Nasional, which is the word for National in the Malay language.

In other words, Petronas is the Malaysian equivalent to British Petroleum or BP, in the UK.
From KL we took an internal flight to Langkawi island where we checked in at the Shangri-la Hotel and Resort.

For each Event, we check to see whether we can derive a Design Pattern based on our Canonical Data Model. If we can, then we have validated our CDM.

3.1 Event 1 - Check-in to a Hotel
Checking-in to the Shangri-la hotel is Event number 1.

As you can see, it is in a beautiful location and consists of over 400 wooden chalets that blend in very naturally with the forest.
3.1.1 The CDM Design Pattern

This Section discusses how the Canonical Data Model (CDM), shown in Section 2.1, applies to the Event of Checking-in to a Hotel. The CDM provides a Design Pattern for the Event-oriented Data Models that we need.

When we think about the Check-in procedure, we realise that in a small transaction, we have obtained a rich source of material for a Data Model:

- Customer name and Address
- Customer Credit card details
- Customer Passport details
- Hotel name and location
- Hotel Room
- Duration of Guest Stay

Using this source data, we can apply the Design Pattern to the Hotel Check-in Event and produce this Data Model as the result (arrows point from Children to Parents):

![Data Model Diagram]

**Business Rules**

It is very good practice to write out the Business Rules that define the conditions that the logic of the Model must comply with. They can then be reviewed and agreed with a Subject Matter Expert (SME).

If it is appropriate, we can use a Business Rules Engine to automate the implementation of the Rules. In this case, the Rules look like this:

- A Customer has one and only one Address.
- A Customer has one or more Addresses.
- A Hotel belongs to one and only one Hotel Chain.
- A Hotel has one and only one Address.
- A Reservation is associated with one Customer.
- A Reservation is associated with one member of Staff.
- A Room belongs to one and only one Hotel.
- A Room Card or Key is associated with one and only one Room and Reservation.
3.1.2 Mapping to the CDM

This Table shows how the Entities in our Hotel Check-In Data Model map on to our Design Pattern based on our Canonical Data Model (CDM).

We are very happy to see that it does because it helps to validate the CDM.

<table>
<thead>
<tr>
<th>CDM</th>
<th>EVENT : Hotel Check-In</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers</td>
<td>Guests</td>
<td></td>
</tr>
<tr>
<td>Documents</td>
<td>Room Card / Key</td>
<td>This the nearest we get to a Document</td>
</tr>
<tr>
<td>Events</td>
<td>Hotel Check-In</td>
<td></td>
</tr>
<tr>
<td>Locations</td>
<td>Hotel Address</td>
<td></td>
</tr>
<tr>
<td>Organisations</td>
<td>Hotel Chains</td>
<td></td>
</tr>
<tr>
<td>Organisations</td>
<td>Hotels</td>
<td></td>
</tr>
<tr>
<td>Organisations</td>
<td>Staff in the Hotel</td>
<td>Staff can be involved and represent the Organisation</td>
</tr>
<tr>
<td>Products or Services</td>
<td>Room, Room Reservation</td>
<td></td>
</tr>
<tr>
<td>Third Parties</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

3.1.3 Message Format

This shows how the Generic Message Template applies to the Specific Hotel Check-In Event. It defines the Source Data for this Event.

<table>
<thead>
<tr>
<th>Generic</th>
<th>Supplier</th>
<th>Date &amp; Time</th>
<th>Customer Details</th>
<th>Products or Services</th>
<th>Unit Price</th>
<th>From Date</th>
<th>To Date</th>
<th>Total Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific</td>
<td>Hotel name</td>
<td>Check-in Date &amp; Time</td>
<td>Barry’s Name, Address and Credit Card Details</td>
<td>Room number</td>
<td>Price per Night</td>
<td>From Date</td>
<td>To Date</td>
<td>---</td>
</tr>
</tbody>
</table>
3.1.4 Data Warehouse for Hotel Check-In

The design of the Data Warehouse is derived from the data in the Check-in Event, which we have also recorded in the Message.

The Data Warehouse will contain the data defined for the Hotel Check-in Event and is essentially an ERD (Entity-Relationship Diagram) version of the Model shown in 3.1.1.

This diagram shows only the names of the Entities in the Data Model.

The attributes are shown in the Dimensional Model in the next Section.

In this diagram, we have positioned the Room Card/Key so that it corresponds to the Document Entity in the CDM.

We have also shown the Hotel_Address differently to reflect a more logical relationship to the Hotel entity, whereas in the CDM, it is shown, for convenience sake, in a Many-to-Many Relationship with the Events entity.

NOTE 1
The one-to-many relationship between Staff and Hotel Reservations is optional at the Staff end because Staff details are not always recorded when a Reservation is made.

NOTE 2
We changed rooms during our stay so this design is necessary to allow for multiple Rooms for one Reservation.
3.1.5 Dimensional Model

Derived data must not appear in an ERD, therefore the Room Card/Key does not appear in this ERD because the data is derived from data already recorded.
4. The BI Layer

4.1 The Approach

Now we come to the Business Intelligence Layer – which is, of course, the reason for everything that has gone before.

Key Performance Indicators ('KPIs') provide an excellent starting-point for Business Intelligence.

We will use some basic KPIs defined for Profitability and Operational Performance.

These can be enhanced in collaboration with the business users to establish commitment.

Profitability can initially be defined as a basic difference between Profits and Revenue.

One example of Performance KPIs is measurement of Deliveries being ‘On-Time’ against plans agreed in Service Level Agreements (SLAs).

In our work we do not have that kind of information available, so we will use a simple measure of the number of Customer Complaints.

The initial Data Architecture will include this basic structure of KPI and Source Data.

This will provide a simple but flexible framework which can be enhanced in a controlled manner.

We will use this as a starting-point for discussions to establish the requirements with the business users.
4.2 BI Discussion

We have adopted a general approach, where we classify Car Hire companies, Hotels, Shops and Tourist Attractions as Brand Chains and specific outlets as Branches.

Therefore a particular Hotel is a Branch and the Branch Chain would be Hilton or Shangri-la.

Here are a few simple examples of the kind of questions that we can answer with this approach.

For the Finance Director of a Hotel chain :-

- Which Hilton Hotel was the most profitable last month ?
- Which Hilton had the most complaints last week as a percentage of total Customers ?

For the Marketing Director of a Car Hire Company :-

- What was the most popular type of car last week ?
- What was the most profitable type of car last week ?

For the Head of Tourism :-

- What was the most popular Tourist Attraction last month ?
- What was the most popular Tourist Attraction over the past six months ?
4.3 Data Marts
We define separate Data Marts for Revenue, Costs and Complaints.

When we consider the implementation options for the BI layer, we have quite a number of options. These include the Microstrategy Personal Cloud, as well as some KPI-oriented Cloud-based offerings.

Firstly, we design our Data Marts and then we have the basis for a selection criteria.
4.4 BI Screenshots
Here we show some examples of Screenshots for currently available Cloud-based BI Products.

1) KPI Library
   • http://benchmark.kpilibrary.com/

2) Smart KPIs - recommended
   • http://www.smartkpis.com/i_kpi/industries/professional-services/

3) Microstrategy Cloud Personal
Here is the Web Site Link (Sign in as barryw@dba.org, with a password of m123 :-
   • http://www.microstrategy.com/cloud/personal/

4) Stacey Barr
Stacey is a prolific writer on KPIs and Performance Measurement :-
   • http://www.staceybarr.com/products/performancemeasureblueprint.html
5. What have we Learned?
We have learned some very valuable lessons, including:

- The importance of a Canonical Data Model
- How to combine Entities where inheritance is involved
- How to load data into a Data Warehouse
- The Steps in loading data from Source Data into a BI Layer

When we should consider using multiple Data Marts with Conformed Dimensions to meet our BI requirements.
6. Here’s your Chance

This Chapter offers you the chance to plan loading new data into our Data Warehouse.

You will do it by completing the model answers, which are shown below as Template documents.

If you feel like sending us your answers we will be happy to give you our comments. You can send them to comments@databaseanswers.org.

6.1 Event: Coffee in Penang Airport

After we checked out of the Shangri-la, we took a limo to Penang Airport for the first leg of our trip back home to London.

With plenty of time we got a coffee at the Coffee Bean, which was started by Herbert Hyman in California in 1963. It is now an international operation, and very popular in Malaysia. Looking for a suitable photo, I came across an excellent one on this page:


![Coffee Bean](image-url)
Here is our Sales Receipt from the Coffee Bean, which we were glad to see validated our CDM:

6.1.1 The CDM Design Pattern
This Section shows the CDM Design Pattern.
Please think about how it applies to the Event of “Stopping for Coffee” and change it accordingly.

[Diagram showing the CDM Design Pattern with nodes labeled: Chains, Supplier, Customers, Products or Services, Event: Stop for Coffee, Staff, Address, Documents]
6.1.2 Mapping to the CDM

This Table is your starting-point for defining how the Entities correspond to the Entities in your ‘Stopping for a Coffee’ Event.

Replace the question marks by your answers.

<table>
<thead>
<tr>
<th>CDM Generic Entities</th>
<th>EVENT : Stop for Coffee Entities</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers</td>
<td>???</td>
<td></td>
</tr>
<tr>
<td>Documents</td>
<td>???</td>
<td></td>
</tr>
<tr>
<td>Events</td>
<td>Stop for Coffee</td>
<td></td>
</tr>
<tr>
<td>Locations</td>
<td>???</td>
<td></td>
</tr>
<tr>
<td>Organisations</td>
<td>???</td>
<td></td>
</tr>
<tr>
<td>Organisations</td>
<td>???</td>
<td></td>
</tr>
<tr>
<td>Organisations</td>
<td>Staff in the Hotel</td>
<td>Staff can be involved and represent the Organisation</td>
</tr>
<tr>
<td>Products or Services</td>
<td>Room Reservation</td>
<td></td>
</tr>
<tr>
<td>Third Parties</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

6.1.3 Message Format

This shows how the Generic Message Template applies to the Specific Hotel Check-In Event.

<table>
<thead>
<tr>
<th>Generic</th>
<th>Supplier</th>
<th>Date &amp; Time</th>
<th>Customer Details</th>
<th>Products or Services</th>
<th>Unit Price</th>
<th>From Date</th>
<th>To Date</th>
<th>Total Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific</td>
<td>Shangri-la Hotel</td>
<td>Check-in Date &amp; Time</td>
<td>Barry’s Name and Credit Card Details</td>
<td>Room number</td>
<td>Price per Night</td>
<td>From Date</td>
<td>To Date</td>
<td>---</td>
</tr>
</tbody>
</table>
6.1.4 Data Warehouse for Hotel Check-In

This is the same as Third-Normal Form Data Warehouse as the one shown in Section 3.1.4.

NOTE 1
The one-to-many relationship between Staff and Hotel Reservations is optional at the Staff end because Staff details are not always recorded when a Reservation is made.

NOTE 2
We changed rooms during our stay so this design is necessary to allow for multiple Rooms for one Reservation.
6.1.5 Dimensional Model

This is the Dimensional Model for the Hotel Check-in Event, which you can use as your starting-point.
Appendix A : Events 1 to 9

A.0 Adding new Data Sources
A very important activity that we have to plan for is adding new Data Sources to our Data Warehouse.

This is often as a direct response to a new business requirement.

The Steps that we will follow are those that we describe in the succession of Events in this Appendix.

A.1 Event 1 - Check-in to a Hotel
Checking-in to the Shangri-la hotel is Event number 1.

A.1.1 The CDM Design Pattern
This Section discusses how the Canonical Data Model (CDM), shown in Section 2.1, applies to the Event of Checking-in to a Hotel. The CDM provides a Design Pattern for the Event-oriented Data Models that we need.

The Design Pattern based on the Hotel Check-in Event looks like this :-

![Diagram of Hotel Check-in Event Design Pattern]

- Hotel Chains
- Customers
- Hotel Reservations
- Staff
- Rooms
- Hotel Address
- Room Card / Key
- Hotels
- Customers
- Hotel Reservations
- Staff
- Rooms
- Hotel Address
- Room Card / Key
- Hotels
A.1.2 Mapping to the CDM

This Table shows how the Entities in our Hotel Check-In Data Model map on to our Design Pattern based on our Canonical Data Model (CDM).

We are very happy to see that it does because it helps to validate the CDM.

<table>
<thead>
<tr>
<th>CDM</th>
<th>EVENT : Hotel Check-In</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers</td>
<td>Guests</td>
<td></td>
</tr>
<tr>
<td>Documents</td>
<td>Room Card / Key</td>
<td>This is as near as we get to a document in a Hotel Check-in</td>
</tr>
<tr>
<td>Events</td>
<td>Rent a Room</td>
<td></td>
</tr>
<tr>
<td>Locations</td>
<td>Hotel Address</td>
<td></td>
</tr>
<tr>
<td>Organisations</td>
<td>Staff in the Office</td>
<td>Staff are usually involved, but not always and represent the Organisation</td>
</tr>
<tr>
<td>Products or Services</td>
<td>Room for Rent</td>
<td></td>
</tr>
<tr>
<td>Third Parties</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

A.1.3 Message Format

This shows how the Generic Message Template applies to the Hotel Check-In Event.

It defines the Source Data for this Event.

<table>
<thead>
<tr>
<th>Generic</th>
<th>Supplier</th>
<th>Date &amp; Time</th>
<th>Customer Details</th>
<th>Products or Services</th>
<th>Unit Price</th>
<th>From Date</th>
<th>To Date</th>
<th>Total Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific</td>
<td>Shangri-la Hotel</td>
<td>Check-in Date &amp; Time</td>
<td>Barry’s Name and Credit Card Details</td>
<td>Room number</td>
<td>Price per Night</td>
<td>From Date</td>
<td>To Date</td>
<td>---</td>
</tr>
</tbody>
</table>
A.1.4 Data Warehouse for Hotel Check-In
At this point, the Data Warehouse will contain data only for the Hotel Check-in Event and therefore it will look like the CDM.

This diagram shows the Entities in the Data Model.

The attributes are shown in the Dimensional Model in the next Section.

In this diagram, we have positioned the Room Card/Key so that it corresponds to the Document Entity in the CDM.

We have also shown the Hotel_Address entity with its correct relationship to the Hotel entity, whereas in the CDM, it is shown, for convenience sake, in a Many-to-Many Relationship with the Events entity.

This is the same as Third-Normal Form Data Warehouse as the one shown in Section 3.1.4.

**Diagram:**
- Hotel_Chains
- Addresses
- Customers
- Credit_Cards
- Hotels
- Hotel_Reservations
- Staff
- Rooms
- Hotel_Reservations_Rooms
- Room Card / Key

**Notes:**
1. The one-to-many relationship between Staff and Hotel Reservations is optional at the Staff end because Staff details are not always recorded when a Reservation is made.
2. We changed rooms during our stay so this design is necessary to allow for multiple Rooms for one Reservation.
A1.5 Dimensional Model

Derived data must not appear in an ERD, therefore the Room Card/Key does not appear in this ERD because the data is derived from data already recorded.
A.2 Event 2 - Hire a Car

A.2.1 The CDM Design Pattern

This Section discusses how the Canonical Data Model (CDM), shown in Section 2.1, applies to the Event of Hiring a Car. The CDM provides a Design Pattern for the Event-oriented Data Models that we need.

The Design Pattern based on the Car Hire Event looks like this:-

A.2.2 Mapping to the CDM

This Section discusses how the Canonical Data Model applies to the Event of Hiring a Car.

We hired a car from a local Car Hire company at the airport when we landed on Langkawi island.

We would normally create a Subject Area Model for Cars, to show details such as Car Make and Model.

This Table shows how the Entities in our Car Hire Data Model map on to our Design Pattern based on our Canonical Data Model (CDM).

We are very happy to see that it does because it helps to validate the CDM.

<table>
<thead>
<tr>
<th>CDM</th>
<th>EVENT : Hire a Car</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers</td>
<td>Customers</td>
<td></td>
</tr>
</tbody>
</table>
### Documents

| Car Hire Contract |

### Event

| Car Hire a Car |

### Locations

| Car Hire Office Address |

### Organisations

| Car Hire Company |

Staff are usually involved and represent the Organisations in the CDM

### Products or Services

| Car Hire |

### Third Parties

| N/A |

#### A.2.3 Message Format

We hired a car from a local rental company in the airport in Langkawi, which worked out very well.

<table>
<thead>
<tr>
<th>Car Hire Company</th>
<th>Date &amp; Time</th>
<th>Customer Details</th>
<th>Products or Services</th>
<th>Unit Price</th>
<th>From Date</th>
<th>To Date</th>
<th>Total Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC Car Hires</td>
<td>Check-in Date &amp; Time</td>
<td>Barry’s Name and Credit Card Details</td>
<td>Car Reg Number</td>
<td>Rental charge per Day</td>
<td>From Date</td>
<td>To Date</td>
<td>---</td>
</tr>
</tbody>
</table>

---
A.2.4 Data Warehouse for Car Hire
At this point, we add the Car Hire data to the data for the Hotel Check-in Event which is already in the Data Warehouse.

A.2.5 Consolidated Data Warehouse
After a little thought, we have combined Car Hire and Hotel Reservations into Suppliers and Services.
A.2.6 Dimensional Model

This is Event 2 where we add the Dimensions and Facts for Hotel Check-in to the existing ones for Car Hire.
A.3 Event 3 - Go Shopping
Harrods Store in the Airport at Kuala Lumpur, Malaysia.

Harrods is very popular in Malaysia, as you can tell from the customers browsing in the store.
A.3.0 Discussion
This Section discusses Sales Receipts, both Generic and Specific.

A.3.0.1 Specific Sales Receipts
Here we have Data Models for Receipts from Harrods, Starbucks and Tesco in Malaysia.

A.3.0.2 Specific Sales Receipts
This shows a consolidated Receipt that provides a generic view of the three specific examples above.
A.3.1 The CDM Design Pattern

This Section discusses how the Canonical Data Model applies to the Shopping Event.

We go Shopping – which is when the long-suffering husband says one of three things :-

1. “It’s a tough job but someone’s got to do it”
2. “When the going gets tough, the tough go shopping”
3. “Yes, dear”

But usually, we survive the experience ;-0)

This is how the CDM Design Pattern applies to the Shopping Event :-
A.3.2 Mapping to the CDM

This Section discusses how the Canonical Data Model applies to the Event of Shopping.

We went shopping at a number of stores in Malaysia.

We would normally create a Subject Area Model for Shopping.

This Table shows how the Entities in our Shopping Data Model map onto our Design Pattern based on our Canonical Data Model (CDM).

We are very happy to see that it does because it helps to validate the CDM.

<table>
<thead>
<tr>
<th>CDM</th>
<th>EVENT : Go Shopping</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers</td>
<td>Customers</td>
<td></td>
</tr>
<tr>
<td>Documents</td>
<td>Sales Receipt</td>
<td></td>
</tr>
<tr>
<td>Events</td>
<td>Go Shopping</td>
<td></td>
</tr>
<tr>
<td>Locations</td>
<td>Stores</td>
<td></td>
</tr>
<tr>
<td>Organisations</td>
<td>Staff, Stores and Store Chains</td>
<td>For example, Harrods and Tesco</td>
</tr>
<tr>
<td>Products or Services</td>
<td>Retail Products</td>
<td></td>
</tr>
<tr>
<td>Third Parties</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

A.3.3 Message Format

The Message Format for this Event will resemble the Sales Receipt.

<table>
<thead>
<tr>
<th>Store Name</th>
<th>Date &amp; Time</th>
<th>Customer Details</th>
<th>Products or Services</th>
<th>Unit Price</th>
<th>From Date</th>
<th>To Date</th>
<th>Total Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harrods</td>
<td>Visit Date &amp; Time</td>
<td>Cash (Anonymous) or Barry’s Name and Credit Card Details</td>
<td>One or many Products</td>
<td>Purchase Price</td>
<td>N/A</td>
<td>N/A</td>
<td>To be calculated</td>
</tr>
</tbody>
</table>
A.3.4 Data Warehouse for Shopping

This shows the Data for the Shopping Event:

![Data warehouse diagram for shopping]

A.3.5 Consolidated Data Warehouse

At this point, we add the Shopping data for Event 3 to the data for the Car Hire and Hotel Check-in Events which is already in the Data Warehouse, so our design looks like this:

![Consolidated data warehouse diagram]

At this point, we would normally consider creating a Glossary of Terms to establish agreed definitions of the word that are in common use.
A.3.6 Dimensional Model

The Dimensional Model will have data for Shopping, Car Hire and Hotel Reservations.
A.4 Event 4 - Visit the Elephant Sanctuary
The Elephant Sanctuary was the first Tourist Attraction that we visited.

Here we see an Elephant family in the Kuala Gandah Sanctuary.

A.4.0 Discussion

A.4.0.1 Elephants
Elephants, especially in small numbers that you see in a Sanctuary or a Circus, frequently have names and we often know their ages.

However, this is not true for crocodiles.

Therefore, we store names and ages for elephants but not for crocodiles so here is the Data Model for Elephants :-
A.4.1 CDM Design Pattern

This Section discusses how the Canonical Data Model applies to the Event of Visiting an Elephant Sanctuary.

We were on Langkawi island where there are a lot of interesting things to see and do.

My wife voted for a trip to Elephant Sanctuary because she thinks baby Elephants are very cute.

So we decided on the Elephant Sanctuary, then the Crocodile Farm and finally the Underwater World.

They have an overhead aquarium and I have always wanted to see fish going over my head, and it had a number of individual Attractions, including the Fish Aquarium and the Penguin Area.

Here is our ticket for the Elephant Ride (called a “Dumbo” Boarding Pass !!!) :-

In our Canonical Data Model (“CDM”) this is an example of a Document related to an Event.

In other words, this is an example of how we are able to validate our CDM.
So we started out trip with a visit to the Elephant Sanctuary :-

A.4.2 Mapping to the CDM
This Section discusses how the Design Pattern for Visiting a Tourist Attraction maps to the Canonical Data Model applies to the Event of Visiting a Tourist Attraction.

If we need to include more detail we would probably create a Subject Area Model for Tourist Attraction.

This Table shows how the Entities in our Tourist Attraction Data Model map on to our Design Pattern based on our Canonical Data Model (CDM).

We are very happy to see that it does because it helps to validate the CDM.

<table>
<thead>
<tr>
<th>CDM</th>
<th>EVENT : Visit to Elephants</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers</td>
<td>Tourists</td>
<td></td>
</tr>
<tr>
<td>Documents</td>
<td>Tickets</td>
<td></td>
</tr>
<tr>
<td>Events</td>
<td>Visit to Elephant Sanctuary</td>
<td>An Elephant Sanctuary is an example of a Tourist Attraction</td>
</tr>
<tr>
<td>Locations</td>
<td>Address of Elephant Sanctuary</td>
<td></td>
</tr>
<tr>
<td>Organisations</td>
<td>Elephant Sanctuary Owners</td>
<td></td>
</tr>
<tr>
<td>Products or Services</td>
<td>Take a Ride on an Elephant</td>
<td></td>
</tr>
<tr>
<td>Third Parties</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>
A.4.3 Message Format
The Message Format will resemble the Tickets.

<table>
<thead>
<tr>
<th>Attraction Name</th>
<th>Date &amp; Time</th>
<th>Customer Details</th>
<th>Services</th>
<th>Unit Price</th>
<th>From Date</th>
<th>To Date</th>
<th>Total Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elephant Sanctuary</td>
<td>Visit Date &amp; Time</td>
<td>N/A</td>
<td>Attraction</td>
<td>Entry Fee</td>
<td>N/A</td>
<td>N/A</td>
<td>As determined</td>
</tr>
</tbody>
</table>

A.4.4 Data Warehouse for the Elephant Sanctuary
This models the Elephant Sanctuary as a Tourist Attraction.
A.4.5 Consolidated Data Warehouse

This consolidates the Tourist Attraction Entities with the existing Entities for Car Hire, Hotel Check-in and Shopping.
A.4.6 Dimensional Model
At this point, data for Event 4 – Elephant Sanctuary (Tourist Attractions) – is added to data for Cars, Hotels and Shopping.
A.5 Event 5 - Visit the Crocodile Farm

I was keen to go to the Crocodile Farm, because I enjoyed the scene in the James Bond “Live and Let Die” movie where he (Roger Moore) escapes from a small island surrounded by crocodiles by jumping across their backs.

I have not been able to find exactly the scene that stuck in my mind, so here is the nearest, which are the crocodiles shown on a movie still :-

In Langkawi we were very impressed to see a brave guy sitting on the back of a crocodile.

Later we found that he was an employee and somehow he had trained the crocodile to let him sit on its back.
A.5.0 Discussion
This Section discusses some of the implications for Data Modeling of combining a Visit to the
Crocodile Farm with a Visit to the Elephant Sanctuary.

A.5.0.1 Adding Crocodiles to Elephant Data Model
Here is the Data Model for a Crocodile.

![Crocodile Data Model](image)

We can see that the Crocodile Entity looks very similar to the Elephant Entity.

The only difference is that we often know the name and age for an Elephant because they are
somehow more ‘user-friendly’ than Crocodiles.

We never know the age and name of a Crocodile !!!

When we try to produce a combined Model for both elephants and crocodiles this is our first draft.
Step 1 : Crocodiles, Elephants and Animals.

This is the first Step to representing a Data Modeller’s view of Crocodiles and Elephants.

This shows that we have created an Animal Entity that has the attributes that Crocodiles and Elephants have in common. Then we have a different Elephant Entity that has only the attributes that are specific to an Elephant. These are Age and Name. Of course, the Age will change every year so when we develop an Entity to be used professionally, we would replace Age by Date of Birth.
Step 2: Crocodiles, Elephants and Animals (part 2)

In this first draft, we use the word Animals to mean both elephants and crocodiles.

At this point, we introduce the concept of Inheritance.

This is shown in our Data Model by the small circle and it means that Elephants inherit the characteristics of Animals and in addition, they have Ages and Names.
Step 3: Crocodiles, Elephants and Species

Of course, we do not normally think of Crocodiles as Animals.

Therefore we have to find another word that can apply equally to both Crocodiles and Elephants.

Species is a word that comes to mind so we will use it, and we will use Sub-Species for specific types of Crocodiles and Elephants.

You can see that we have established that ages and names are usually available to Elephants but not for other species.
A.5.1 The CDM Design Pattern
This Section discusses how the Canonical Data Model applies to the Event of Visiting a Crocodile Farm.

We would expect this to be identical to a Visit to the Elephant Sanctuary.

But it is worth the effort of compiling the Mapping Analysis so that we can double-check the situation.

Sure enough, after we complete the Mapping, we can see that the logic is identical.

Therefore we do not need to change the Design Pattern or the Data Warehouse.

The Dimensional Model will simply have additional data for the Crocodile Farm.

A.5.2 Mapping to the CDM
This Section discusses how the Canonical Data Model applies to the Event of Visiting a Crocodile Farm.

We can see that the Data Model for Crocodile Farm is identical to that for the Elephant Sanctuary.

We can simply create a Data Model for Tourist Attraction and create Event Types of Visits to a Crocodile Farm and an Elephant Sanctuary.

In other words, we handle Elephant Sanctuaries and Crocodile Farms as different sorts of Reference Data.

This Table shows how the Entities in our Crocodile Farm Data Model map on to our Design Pattern based on our Canonical Data Model (CDM).

We are very happy to see that it does because it helps to validate the CDM.

<table>
<thead>
<tr>
<th>CDM</th>
<th>EVENT : Visit to Crocodiles</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers</td>
<td>Tourists</td>
<td></td>
</tr>
<tr>
<td>Documents</td>
<td>Tickets</td>
<td></td>
</tr>
<tr>
<td>Events</td>
<td>Visit to Crocodile Farm</td>
<td></td>
</tr>
<tr>
<td>Locations</td>
<td>Address of Crocodile Farm</td>
<td></td>
</tr>
<tr>
<td>Organisations</td>
<td>Crocodile Farm Owners</td>
<td></td>
</tr>
<tr>
<td>Products or Services</td>
<td>See the Crocodile Farm</td>
<td></td>
</tr>
</tbody>
</table>
A.5.3 Message Format

The Message Format will resemble the Tickets.

<table>
<thead>
<tr>
<th>Attraction Name</th>
<th>Date &amp; Time</th>
<th>Customer Details</th>
<th>Services</th>
<th>Unit Price</th>
<th>From Date</th>
<th>To Date</th>
<th>Total Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>eg Crocodile Farm</td>
<td>Visit Date &amp; Time</td>
<td>N/A</td>
<td>Attraction</td>
<td>Entry Fee</td>
<td>N/A</td>
<td>N/A</td>
<td>As determined</td>
</tr>
</tbody>
</table>
A.5.4 Data Warehouse for the Elephant Sanctuary

The Data Model the Crocodile Farm is logically identical to the Elephant Sanctuary Model.

Therefore we can use the same Model for Tourist Attractions and simply add values to the Reference Data.
A.5.5 Consolidated Data Warehouse

When we review this Data Model, we can see that the logic of the Elephant Sanctuary applies equally to the Crocodile Farm.

In other words, our Consolidated Data Warehouse is identical, and we simply add to the Reference Data as another kind of Tourist Attraction.
A.5.6 Dimensional Model

When we review the Dimensional Model, we can see that we can accommodate Crocodile Farms by simply adding to the Reference Data.
A.6 Event 6 - Visit the Underwater World

The Underwater World was very difficult to find, but when we did find it, it was definitely worth it.

We paid a fee to enter but when we got inside we discovered that there many Attractions inside a very large building,

Entry to these Attractions, like the Aquarium, was free.

After a little thought, I decided to treat the Underwater World as a Tourist Attraction.

Then I decided to treat the Aquarium and Penguins as Tourist Attractions with a ‘Parent’ relationship to the Underwater World.

This is very easy to implement in the Model and makes it very easy to ‘read’ the Model.

This is important because every Data Model should tell a story and therefore a good test of a well-designed Model is that it is easy to read it like a story.

Therefore, the Aquarium and the Penguins both become Attractions with a zero entry fee and have a relationship to the Underwater World as a ‘Parent’.

Here’s a photo to give you a glimpse of the fascinating interior of Underwater World from this page on their Web Site :-

A.7 Event 7 - Visit the Aquarium

Visiting the Aquarium was great and allowed me to tick off another “once-in-a-lifetime” experience which was to walk inside an aquarium looking up at the fish swimming by over my head !!!

This photo shows the view I saw when I looked up and saw fish swimming past above my head.

A.7.1 Mapping to the CDM

From a Data Modelling point of view, visiting an Aquarium is identical to visiting a Crocodile Farm or Elephant Sanctuary.

Therefore we do not need a separate CDM, and we go through the process of mapping simply to confirm that it is identical.

This Table shows how the Entities in our Visit the Aquarium Event map on to our Design Pattern based on our Canonical Data Model (CDM).

<table>
<thead>
<tr>
<th>CDM</th>
<th>EVENT : Visit to Aquarium</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers</td>
<td>Tourists</td>
<td></td>
</tr>
<tr>
<td>Documents</td>
<td>Tickets</td>
<td></td>
</tr>
<tr>
<td>Events</td>
<td>Visit to Aquarium</td>
<td></td>
</tr>
<tr>
<td>Locations</td>
<td>Address of Aquarium</td>
<td></td>
</tr>
<tr>
<td>Organisations</td>
<td>Aquarium Owners</td>
<td></td>
</tr>
<tr>
<td>Services</td>
<td>See the Aquarium</td>
<td></td>
</tr>
<tr>
<td>Third Parties</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>
A.8 Event 8 - Visit the Penguins
The Penguins were also at the Underwater World and I was very lucky again because I saw live penguins which was another ‘first’ for me.

They were a wonderful sight.

It was very hot outside (over 90 degrees in the shade) outside in tropical Malaysia.

But inside we saw Penguins in a very comfortable (for them) sub-zero temperature.
A.8.0 Validation of the Species Data Model

Validation proves that our existing Species Model is correct.

Fish do not have names and their characteristics are the same as Crocodiles.

We can say the same for Penguins.

But Fish and Penguins both have Sub-Species !!!

Therefore we have validated our Species Model and it is good as it stands for Fish and Penguins, as well as Crocodiles and Elephants.

To make it clear that we are talking about the examples of Species that we see in the Underwater World we call the Entity ‘Species in Attraction’.

It would be sensible for us to plan for future Species which might not be a Sub-Species.

The simplest way to do this is to add a self-referencing Relationship to the Species Entity.

This means that the Species can point to itself so that Species, Sub-Species are stored in the same table in a Database.

In HR terms, this is like having an Employee table and saying ‘Each Employee reports to a Superior’.
A.8.1 CDM Design Pattern
This Section shows how the Design Pattern looks for Penguins.

A.8.2 Mapping to the CDM
This Section discusses how the CDM applies to the Penguin Area.

When we check our CDM we can see that it applies in an identical way that it does to Elephants and Crocodiles.

That is we pay for a Service and receive a Document, in the form of a ticket, that allows us to enter the Attraction.

Therefore we do not need a separate CDM, and we go through the process of mapping simply to confirm that it is identical.

This Table shows how the Entities in our Tourist Attraction Data Model map on to our Design Pattern based on our Canonical Data Model (CDM).

<table>
<thead>
<tr>
<th>CDM</th>
<th>EVENT : Visit to Penguins</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers</td>
<td>Tourists</td>
<td></td>
</tr>
<tr>
<td>Documents</td>
<td>Tickets</td>
<td></td>
</tr>
<tr>
<td>Events</td>
<td>Visit to Penguins</td>
<td></td>
</tr>
<tr>
<td>Locations</td>
<td>Address of Penguins Area</td>
<td></td>
</tr>
<tr>
<td>Organisations</td>
<td>Penguin Area Owners</td>
<td></td>
</tr>
<tr>
<td>Services</td>
<td>See the Penguins</td>
<td></td>
</tr>
<tr>
<td>Third Parties</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>
A.8.3 Message Format

Entry to the Penguin Area is free and not monitored after you buy your ticket for the Underwater World and enter.

The Message Format will be simply the one for the Ticket to the Underwater World because there is no separate ticket for the Penguin Area.

<table>
<thead>
<tr>
<th>Attraction Name</th>
<th>Date &amp; Time</th>
<th>Customer Details</th>
<th>Services</th>
<th>Unit Price</th>
<th>From Date</th>
<th>To Date</th>
<th>Total Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Underwater World Visit Date &amp; Time</td>
<td>N/A Penguin Area</td>
<td>Free</td>
<td>N/A N/A</td>
<td>Free</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A.8.4 Data Warehouse for Tourist Attractions
This diagram shows the Design Pattern of the Canonical Data Model adapted for Tourist Attractions.

A.8.5 Dimensional Data Model
This diagram shows the Design Pattern of the Canonical Data Model adapted for Tourist Attractions.
A.9 Event 9 - Check-out from a Hotel

The last Hotel we stayed in was the Shangri-la in Penang.

This was the beautiful view from our room that we were very sad to see for the last time before we checked out:-

![View from the room](image-url)
A.9.1 The CDM Design Pattern

The Credit Card that I use was, of course, associated with me, but it was also associated with payment of the Hotel Bill so there is a relationship between the Credit Card and the Hotel Guest (i.e. Customer) and between the Credit Card and the Total Hotel Bill.

This Section discusses how the Canonical Data Model (CDM), shown in Section 2.1, applies to the Event of Checking-in to a Hotel. The CDM provides a Design Pattern for the Event-oriented Data Models that we need.

The Design Pattern based on the Hotel Check-in Event looks like this:

---

A.9.2 Mapping to the CDM

This Table shows how the Entities in our Hotel Check-out Data Model map on to our Design Pattern based on our Canonical Data Model (CDM).

We are very happy to see that it does because it helps to validate the CDM.

<table>
<thead>
<tr>
<th>CDM</th>
<th>EVENT : Hotel Check-out</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers</td>
<td>Guest</td>
<td></td>
</tr>
<tr>
<td>Documents</td>
<td>Receipts</td>
<td></td>
</tr>
<tr>
<td>Events</td>
<td>Rent a Room, Use Restaurants, Laundry, and so on.</td>
<td></td>
</tr>
<tr>
<td>Locations</td>
<td>Hotel Address</td>
<td></td>
</tr>
<tr>
<td>Organisations</td>
<td>Staff in the Hotel</td>
<td>Staff are always involved in the</td>
</tr>
</tbody>
</table>
A.9.3 Message Format

This shows how the Generic Message Template applies to the Hotel Check-Out Event.

It defines the Source Data for this Event.

<table>
<thead>
<tr>
<th>Generic</th>
<th>Supplier</th>
<th>Date &amp; Time</th>
<th>Customer Details</th>
<th>Products or Services</th>
<th>Unit Price</th>
<th>From Date</th>
<th>To Date</th>
<th>Total Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific</td>
<td>Shangri-la Hotel</td>
<td>Check-in Date &amp; Time</td>
<td>Barry’s Name and Credit Card Details</td>
<td>Room, Meal, Services</td>
<td>Price per Night</td>
<td>From Date</td>
<td>To Date</td>
<td>Derived</td>
</tr>
</tbody>
</table>

A.9.4 Data Warehouse for Hotel Check-Out

At this point, we add specific Check-out data to the Data Warehouse.

This diagram shows the Entities in the Data Model.

The attributes are shown in the Dimensional Model in the next Section.

We have also shown the Hotel_Address entity with its correct relationship to the Hotel entity, whereas in the CDM, it is shown, for the sake of convenience, in a Many-to-Many Relationship with the Events entity.
A.9.5 Consolidated Data Warehouse

At this point, we add the Shopping data for Event 3 to the data for the Car Hire and Hotel Check-in Events which is already in the Data Warehouse, so our design looks like this :-
A.9.6 Dimensional Model

At this point, the Dimensional Model will have the complete set of Dimensions and Facts.

This Data Model shows them all consolidated into a single Dimensional Model.

An alternative design with more than one Fact Table is shown in the BI discussion.

This is normally called multiple Data Marts which require Conformed Dimensions.
A.10 Event: Shipping a Car from the UK to Malaysia

We had such a great time in Malaysia, we decided to move out there for a while. One of the things we had to arrange was to ship our car out. We decided to use Maersk to help us.

Here is the kind of Container Ship we had in mind:

A.10.1 The CDM Design Pattern

This Section shows how the CDM Design Pattern applies to the Event of “Ship a Car to Malaysia”.
A.10.2 Mapping to the CDM
This Table is your starting-point for defining how the Entities correspond to the Entities in your ‘Stopping for a Coffee’ Event.

Replace the question marks by your answers.

<table>
<thead>
<tr>
<th>CDM Generic Entities</th>
<th>EVENT : Ship a Car by Sea</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers</td>
<td>Customer</td>
<td></td>
</tr>
<tr>
<td>Documents</td>
<td>Contracts</td>
<td></td>
</tr>
<tr>
<td>Events</td>
<td>Ship a Car by Sea</td>
<td></td>
</tr>
<tr>
<td>Locations</td>
<td>Booking Office</td>
<td></td>
</tr>
<tr>
<td>Organisations</td>
<td>Maersk</td>
<td></td>
</tr>
<tr>
<td>Organisations</td>
<td>Staff in the Office</td>
<td>Staff can be involved and represent the Organisation</td>
</tr>
<tr>
<td>Products or Services</td>
<td>Shipping Service</td>
<td></td>
</tr>
<tr>
<td>Third Parties</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

A.10.3 Message Format
This shows how the Generic Message Template applies to the Specific Hotel Check-In Event.

<table>
<thead>
<tr>
<th>Generic</th>
<th>Supplier</th>
<th>Date &amp; Time</th>
<th>Customer Details</th>
<th>Products or Services</th>
<th>Unit Price</th>
<th>From Date</th>
<th>To Date</th>
<th>Total Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific</td>
<td>Maersk Line</td>
<td>Check-in Date &amp; Time</td>
<td>Barry’s Name and Credit Card Details</td>
<td>Shipping Service</td>
<td>Quoted price to ship Cargo</td>
<td>From Date</td>
<td>To Date</td>
<td>---</td>
</tr>
</tbody>
</table>
A.10.4 Data Warehouse for Shipping a Car

At first, this Event seems quite different from the others, but after a little thought, we realise that it has a Start and End Date.

In addition, there is a Customer and we are using a Service provided by a Supplier.

Therefore Shipping a Car is therefore logically similar to a stay in a hotel and our Data Warehouse looks like this :-

In general, we might be shipping many things and not just one thing, so our Data Model needs to provide for this. We will call these ‘things’ Commodities.

We can do this by simply start with a many-to-many relationship between Commodities entity and the Shipments entity and resolve it to two one-many relationships, so that our Model looks like this :-
When we think about how we would show this Data Model in a generic format, this is the result:

**A.10.5 Consolidated Data Warehouse**

At this point, we add the Car Shipping data to the data for all the other Events which is already in the Data Warehouse, so our design looks like this:
A.10.6 Dimensional Model

This is the Dimensional Model for the Car Shipping Event.
Appendix B : Data Model Review
This Section tries to describe the analysis and thought processes of a Data Modeller as he or she goes about his work.

Of course, it is subjective and these are my thoughts - that is, Barry Williams.

However, I have been doing this for over 15 years and I find it consistently fascinating.

The reasons I find Data Modelling so fascinating are that it is visual, creative and can be a great help when used as a source of Data Models as a vehicle for communication between business users, management and data management professionals, analysts and developers.

So here we go ;-)0)

B.1 What are we going to do?
We are going to step through the process of designing Data Models and then reviewing the results to see whether we can simply the overall structure by generalising the Entities that we have produced.

It is one of the situations where it takes longer to describe it (which is boring) than doing it (which is interesting).

So (once again !!!) here we go …

B.2 Where did we start?
Our Canonical Data Model is Event-oriented and we started with nine Events :-

1. Check-in a Hotel
2. Hire a Car
3. Go Shopping
4. Visit an Elephant Sanctuary Sanctuary
5. Visit a Crocodile Farms
6. Visit the Unde4rwater World
7. Visit an overhead Aquarium
8. Visit a Penguin Area
9. Check-out of a Hotel
B.3 First Data Model

We have 9 different types of Events and our first Data Model looks like this:

- **Event Types**
  - Aquariums
  - Cars
  - Crocodiles
  - Elephants
  - Hotels
  - Penguins
  - Stores
  - Underwater_World

Note that everything is listed alphabetically. This is good practice because it makes it easier for us to keep track when we have many things to deal with.

Event Types include:
1. Go Shopping
2. Hire a Car
3. Rent a Hotel Room
4. Visit an Aquarium
5. Visit a Crocodile Farm
6. Visit an Elephant Sanctuary
7. Visit the Penguins
8. Visit the Underwater World
B.4 Second Data Model

When we think about the Entities, it seems intuitive that visiting a Crocodile Farm must be similar to visiting an Elephant Sanctuary.

Then we can review the discussion in the Section above to convince ourselves to treat them the same way.
B.5 Third Data Model

At this stage, we are thinking about the underlying relationships between the entities that we have determined are in the scope of our study.

We have started to group entities together in a way that helps us to reduce the overall size of the Data Model by identifying generic characteristics in the specific entities.

However, every Data Model should tell a story and now we have to consider how our story can be told.

For example, we would say ‘Products are found in Stores’ and therefore we should show a one-to-many relationship between Stores and Products.

We can see that it is the other way round in Version 3 because of the way it had evolved so now we have to change it to make it correct.
B.6 Fourth Data Model

At this stage, we start thinking about the story we want to tell and how we should structure the Data Model to reflect the story.

For example, we would say ‘Products are found in Stores’ and therefore we should show a one-to-many relationship between Stores and Products.

At this point, we like the left-hand side of the Data Model and we can say :-

1. There is a ‘Go Shopping’ Event which means buying Products, rather than go to Stores.

2. This allows us to show Products at the same level as Stores which means that we can maintain consistency with the terminology we have used elsewhere in this document.

3. We have decided that Services included “Hiring a Car” and “Staying in a Hotel”.

So now we turn to the right-hand side of the Data Model.
B.7 Fifth Data Model
At this stage, we start thinking about the story we want to tell and how we should structure the Data

On the right-hand side, we show Species as a Tourist Attraction, whereas in fact, people say ‘Let’s go the Elephant Sanctuary’ so the entity that should be related to the Tourist Attraction is the Elephant Sanctuary, rather than the Elephant.

Of course, people can also say “Let’s go and see the Elephants” but in the physical world, the reality is that Elephants are housed in an Elephant Sanctuary and it the Sanctuary that they actually go and see.

So we conclude that it is OK to show the Tourist Attractions and their relationships.

We still have one or two questions to resolve but we are quite content because the Model looks good and the overall logic is good.

B.8 Conclusion
We hope you have found this discussion interesting and useful.
If you have any comments, suggestions or questions, please feel free to email us at :-
  • barry@databaseanswers.org.
Appendix C: Slideshow

This Section shows what is in the Slideshow on our Web Site:


If you would like to take a look at the Slide Show, here is the page on the Database Answers Web Site:

## Appendix D: Glossary of Terms

A Glossary of Terms is very important.

It can establish a commonly accepted meaning for all Terms that are in common usage.

This table shows a simple example.

<table>
<thead>
<tr>
<th>TERM</th>
<th>MEANING</th>
<th>GENERIC EQUIVALENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer</td>
<td>A person or organisation that purchases a Product or Service</td>
<td></td>
</tr>
<tr>
<td>Hotel Guest</td>
<td>Customer</td>
<td>Customer</td>
</tr>
<tr>
<td>Product</td>
<td>Something tangible that can be purchased and physical carried away.</td>
<td></td>
</tr>
<tr>
<td>Service</td>
<td>Something that is provided to a Customer but is not tangible</td>
<td>For example, an Elephant Ride or a Meal.</td>
</tr>
<tr>
<td>Tourist</td>
<td></td>
<td>Customer</td>
</tr>
</tbody>
</table>