4. Design Patterns

4.0 Welcome

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

- http://www.databaseanswers.org/data_models/index.htm

You can also join our Database Answers Community

- http://databaseanswers.ning.com/

Barry Williams

Principal Consultant

Database Answers Ltd.

barryw@databaseanswers.org

4.1 Introduction

This Chapter will discuss the importance of Design Patterns and give some examples.

4.1.1 What is this ?

Design Patterns are solutions to common problems.

4.1.2 Why is it important ?

They are important because they occur all around us.

Design Patterns exist because they reflect the existence of common problems and common solutions in the world of Data Modelling.

Because of this, when we want to understand a Data Model, an obvious starting-point is to look for common Design Patterns.

This document can help in looking at a Data Model and understanding it.

It is based on the concept of Design Patterns which are general solutions to common problems which occur on a regular basis.

This document starts with some simple Concepts and then discusses common Applications that use these Concepts.
This applies in two situations :-

- Data Models created by Reverse Engineering existing Databases.
- Other Data Models.

This document will help in the Quality Assurance ('QA') of these Data Models, which might be produced internally or externally, by Partners, for activities such as Data Migration.

i) For the first situation, it is not appropriate to try to do a Quality Assurance of the Model.
This is primarily because Databases in operational systems have usually gone through a series of changes and the impact on design has not been thought through.
As a result there has not been the knowledge, commitment or time to redesign the Database.
The objective is primarily to understand the Database.

The many-to-many Pattern will not occur because this cannot be implemented directly in a Relational Database. This applies also to Inheritance which can only be identified by implication.

It is often useful to create a general Business Data Model that renames Tables as appropriate to replace the physical Table names with corresponding Business Terms.

For complex Databases, it is usually valuable to create a Top-Level Data Model with lower-level Subject Area Models.

It is important to try to establish a Glossary of Terms, covering descriptions of the most important Tables and Attributes and Reference Data.
Another important activity is to establish the Business Rules that define the logic underlying any Database.
Some simple examples that can be used as Templates have been shown in this document.

ii) For the second situation, it is appropriate to do a Quality Assurance of the Model
This would include a number of tasks, such as :-

- looking for examples of the Design Patterns being used where appropriate.
- review of the Reference Data

4.1.3 What will I Learn ?

You will be able to recognise Design Patterns when they occur and find it much easier to cope with complex situations because you will be able to break them down into simple components.

4.2 Addresses

This is taken from the Data Model shown on this page :-

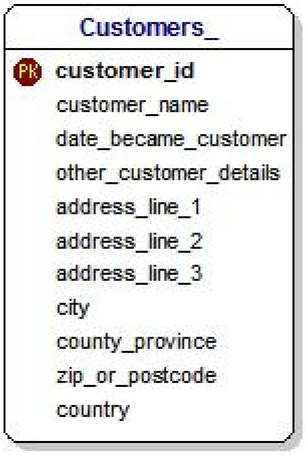
- http://www.databaseanswers.org/data_models/customers_and_addresses/index.htm

There are several Options for modelling Addresses.

Option 1 : The simplest Design

This is the simple and most basic Design.

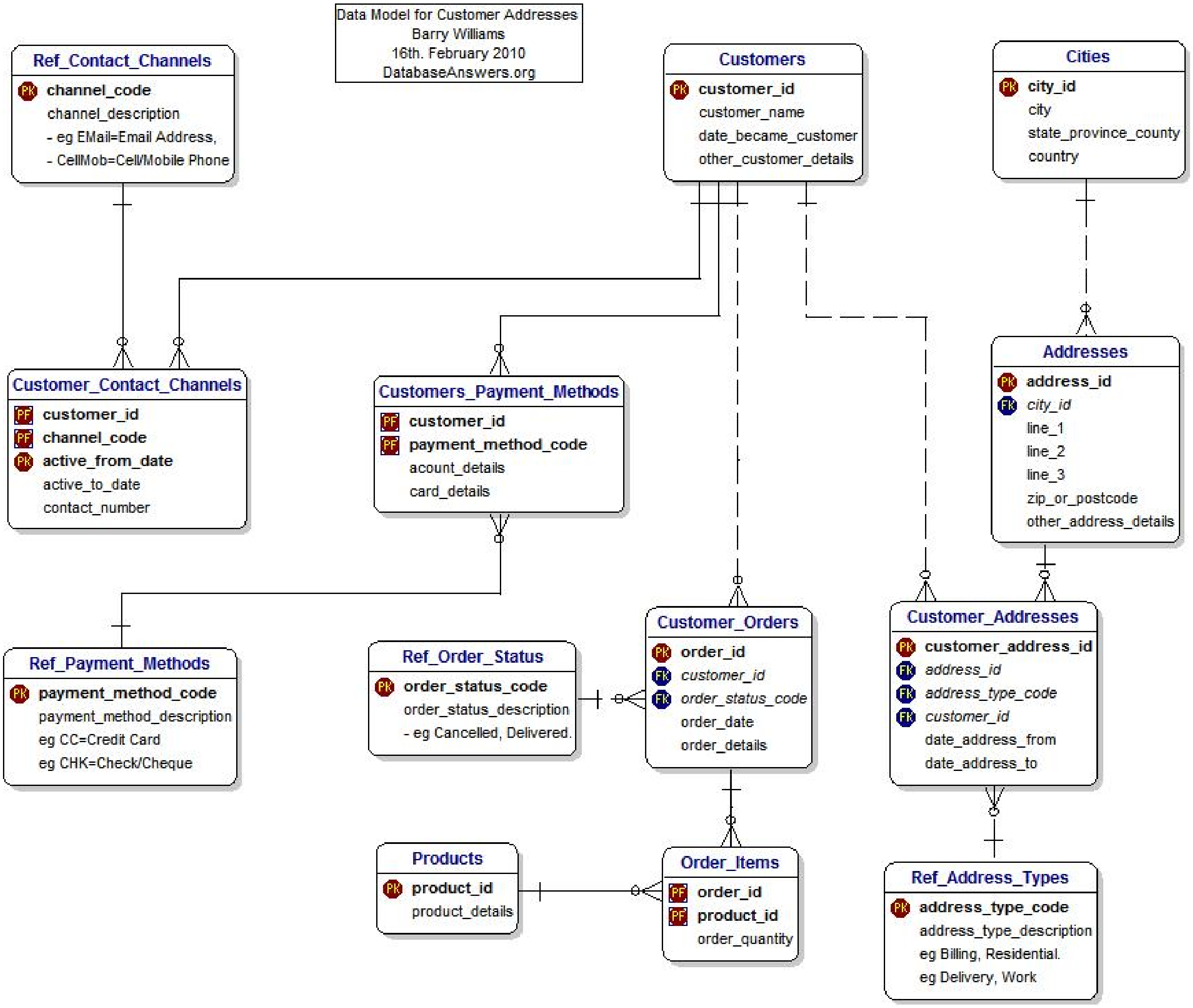
It stores the Address in the Customer Table :-



Option 2 : A separate Address Table

This Design stores the Address in a separate Table, as shown in this example :-

- http://www.databaseanswers.org/data_models/customers_and_addresses/index.htm



Option 3 : A more general Standards-based Design

This Design also stores the Address in a separate Table which then makes it possible to validate Addresses using bespoke or commercial software, such as QAS :-

- In the US - http://www.qas.com/
- In the UK - http://www.qas.co.uk/

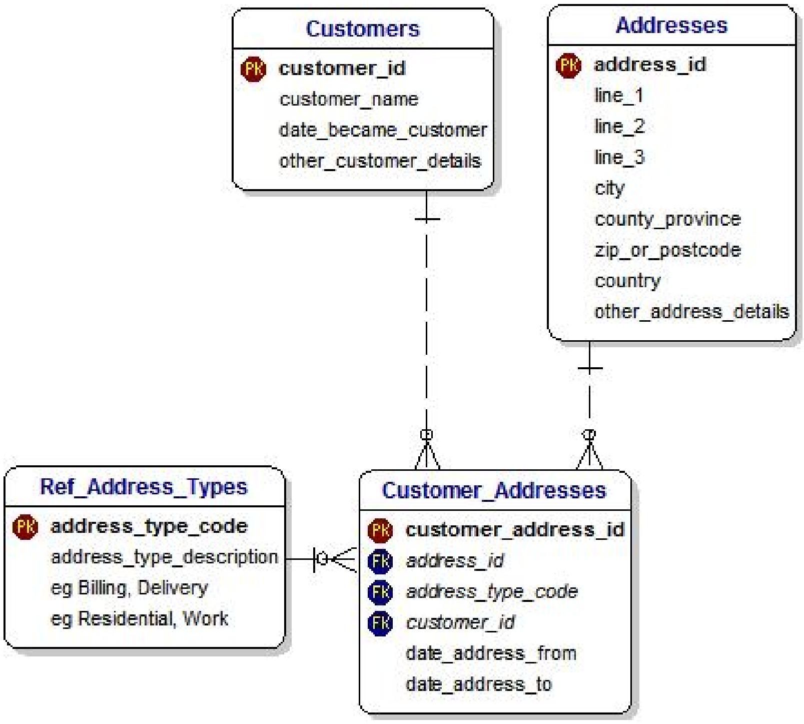
There is a two-line standard recommended by the US Postal Service :-

- http://pe.usps.com/text/pub28/PUB28C2.html
- http://pe.usps.com/

You can find a discussion of the Universal Postal Union UPU S42 International Address Standard, published in 2003 by following this Link :-

- http://xml.coverpages.org/Lubenow-UPUS42.html

The UK standard called the 'PAF File' or 'Post Office Address File' , which favours 4 lines. This design is compatible with the US and the UK standard :-



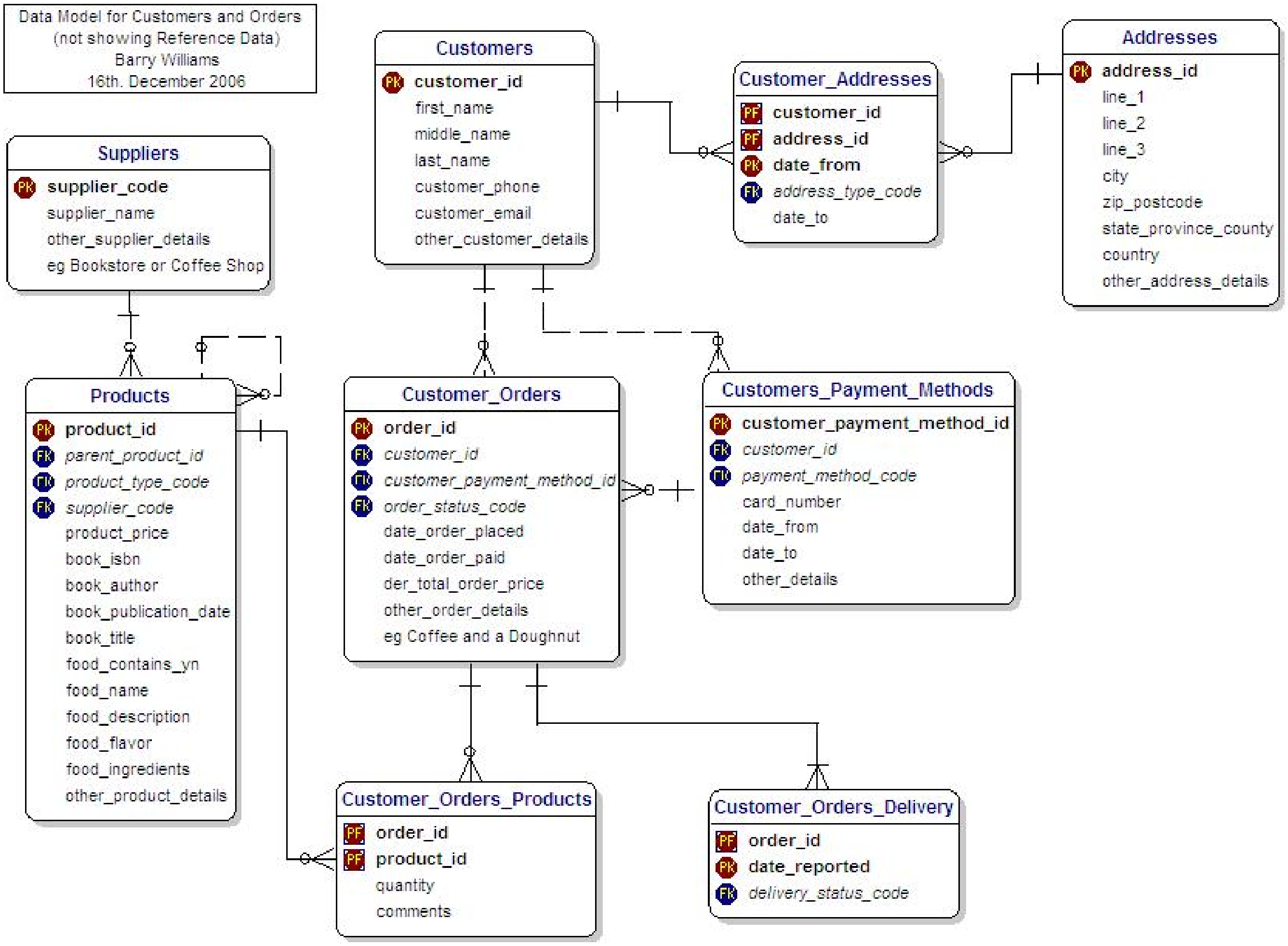
4.3 Master-Detail

Master-Detail is a very common design and is described on thi page on Wikipedia :-

- http://en.wikipedia.org/wiki/Master-detail

A typical example Master-Detail is Customers and Orders :-

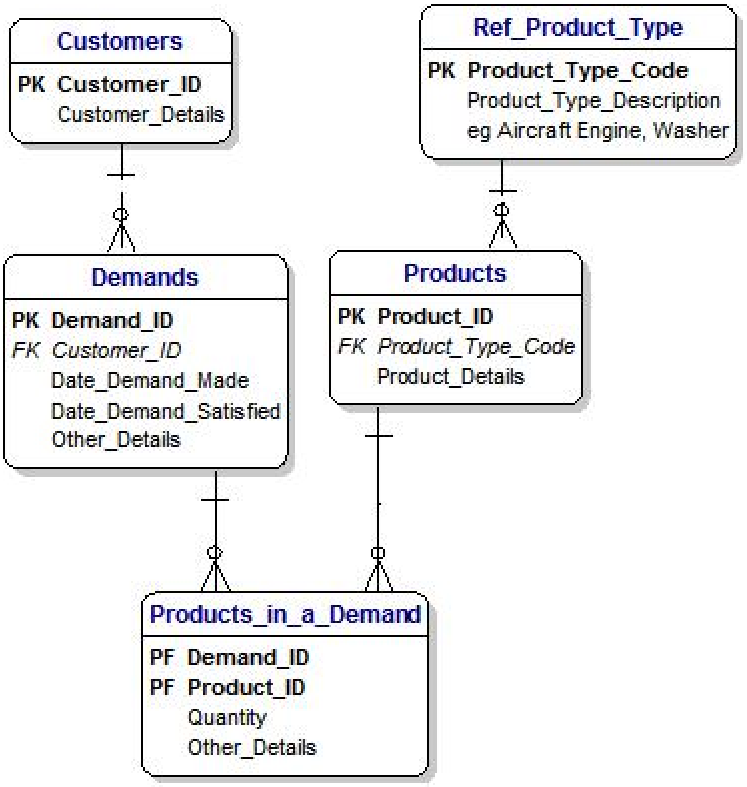
- http://www.databaseanswers.org/data_models/customers_and_orders/index.htm



4.4 Data Warehouses

4.4.1 Design of an ERD

This Data Model is an Entity-Relationship-Diagram ('ERD') for Customers and Demands :-



We could describe it in these terms :-

> "Customers place Demands for Products of different Types."

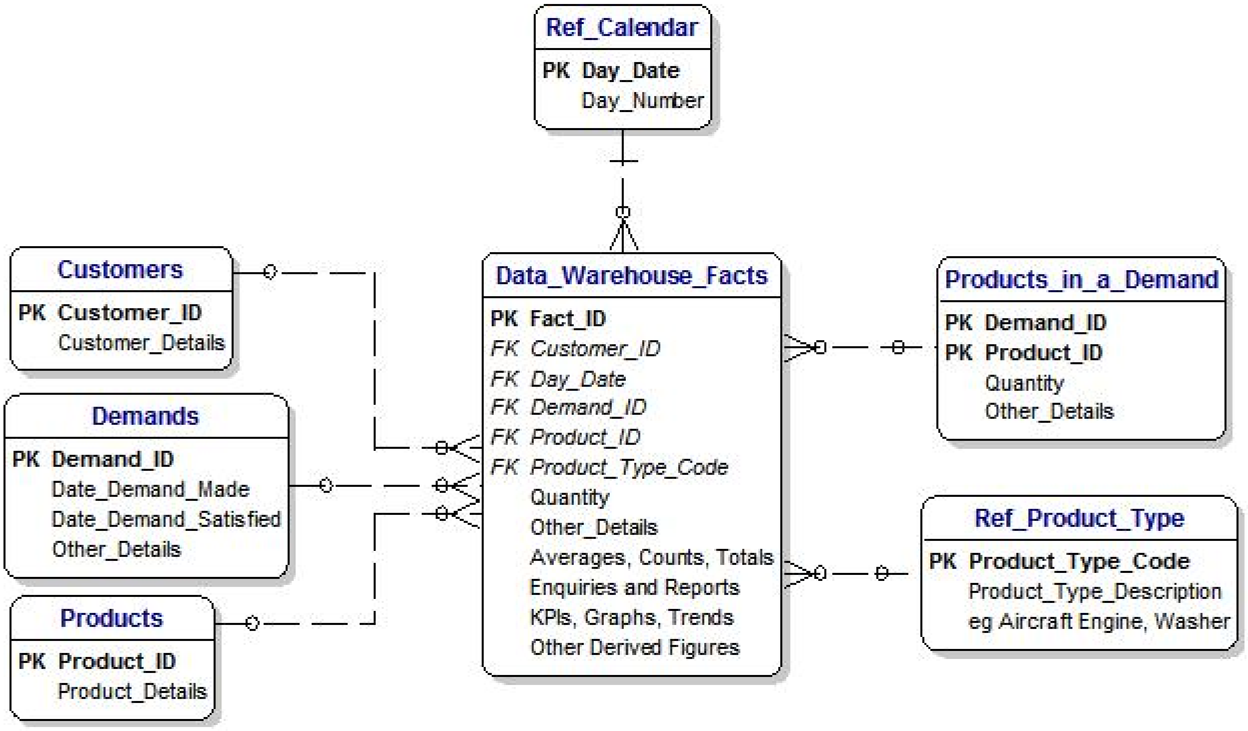
4.4.2 Design of a Data Warehouse

For our purposes, we can consider a Data Warehouse to be the same as a Data Mart.

In our professional experience, we have designed Data Marts which had a specific scope and timescale and defined Users.

A Data Warehouse, on the other hand, simply puts all data into a 'big basket' and says 'Here it is, come and get it'.

This Data Model shows the corresponding Data Warehouse for Customers and Demands :-



4.4.3 Reviewing the Design of a Data Warehouse

The design of any Data Warehouse will conform to this patter with Dimensions and Facts. Dimensions correspond to Primary Keys in all the associated Tables (ie the Entities in the ERD)   and the Facts are the derived values that are available.

Therefore, reviewing the Design of a Data Warehouse involves looking for this Design Pattern.

With one exception, the Relationships are optional because the Enquiries need not involve any particular Dimension.

The one exception to this rule is that the Relationship to the Calendar is mandatory because an Enquiry will always include a Date.
Of course, an Enquiry might include all data since the first records, but the principle still applies.

The purpose of the Data Warehouse is to make it easy to retrieve data in any combination in order to answer questions like this :-

- Which Customers ordered the most Products ?
- What was the most popular Product in the first week of April ?
- What was the average time to respond to demands for the most popular Product ?
- How many Demands did we receive in May ?

4.5 Applications

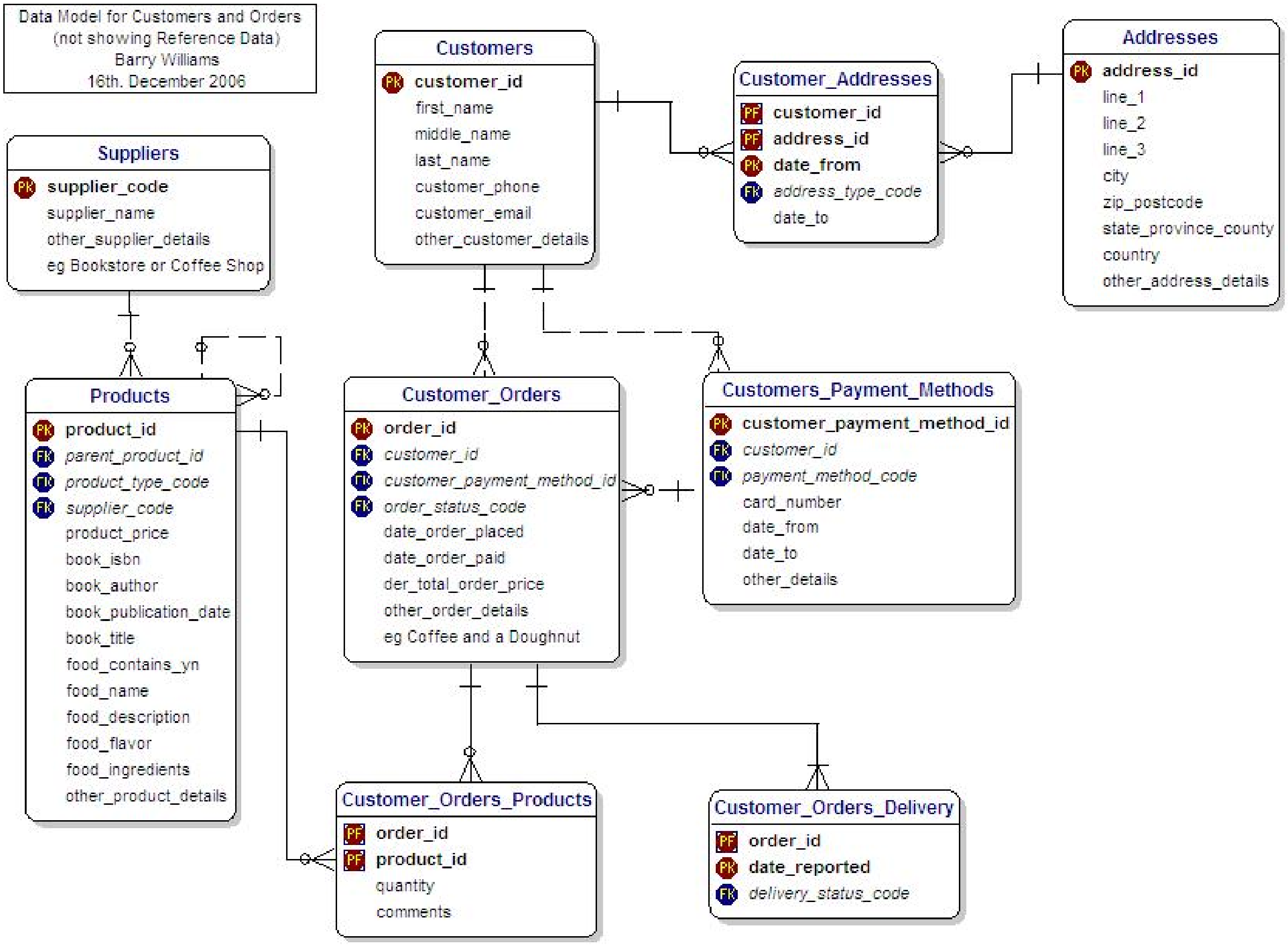
This shows some examples of simple Applications that occur frequently.

4.4.1 Customers and Orders

This typical Customers and Orders Data Model represents a widespread kind of application :-

- http://www.databaseanswers.org/data_models/customers_and_orders/index.htm

It is an example of how a Many-To-Many Relationship between Customer Orders and Products is broken down into two One-to-Many Relationships.
One of these is Customers-_Orders to Customer-Order-Products and the other is Products to Customers_Order_Products.



4.4.2 Deliveries – Simple

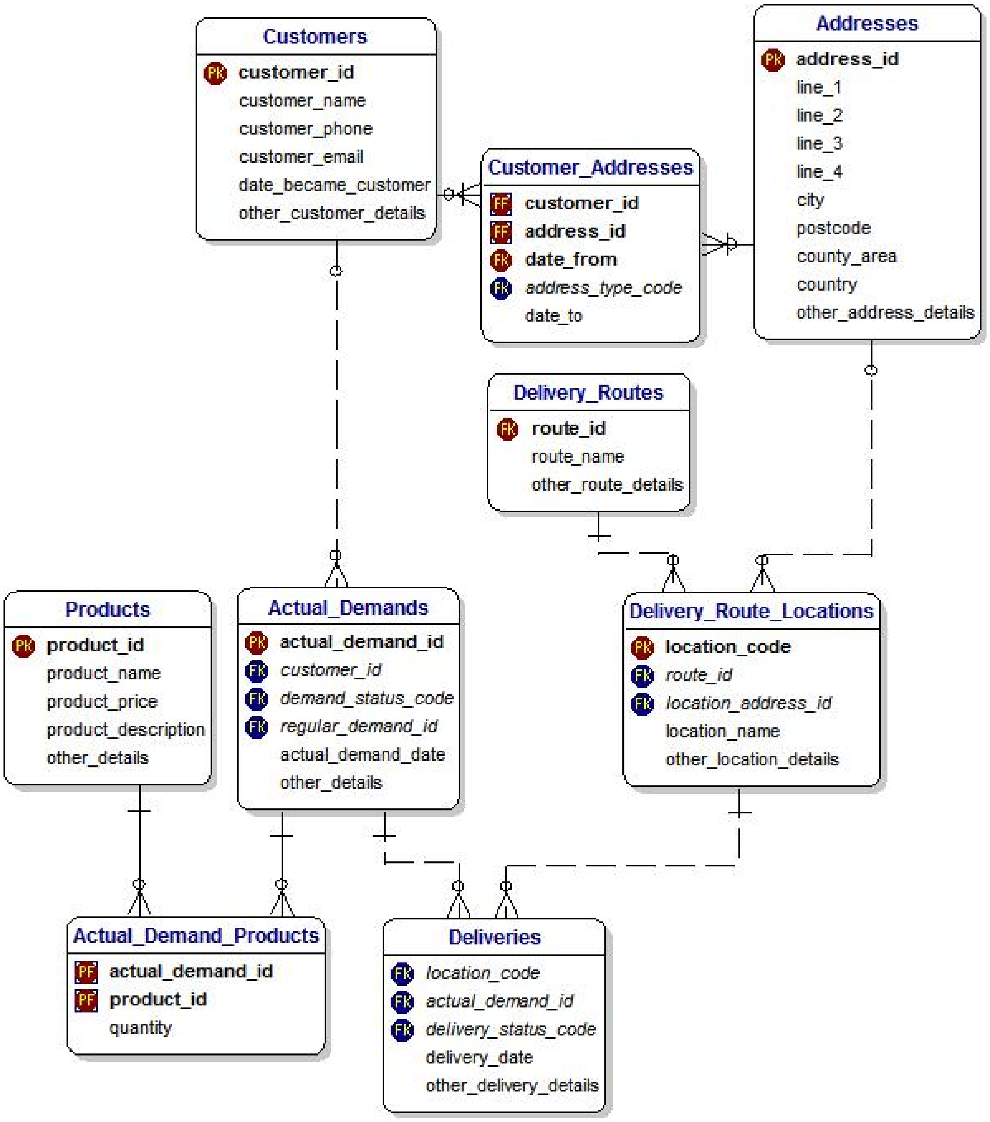
This Data Model is a simple Design Pattern that covers the activities of **One-off Deliveries** to a designated address.

The process of reviewing a Data Model is to ask :-

“How do I describe the Business Rules behind this Model ?”

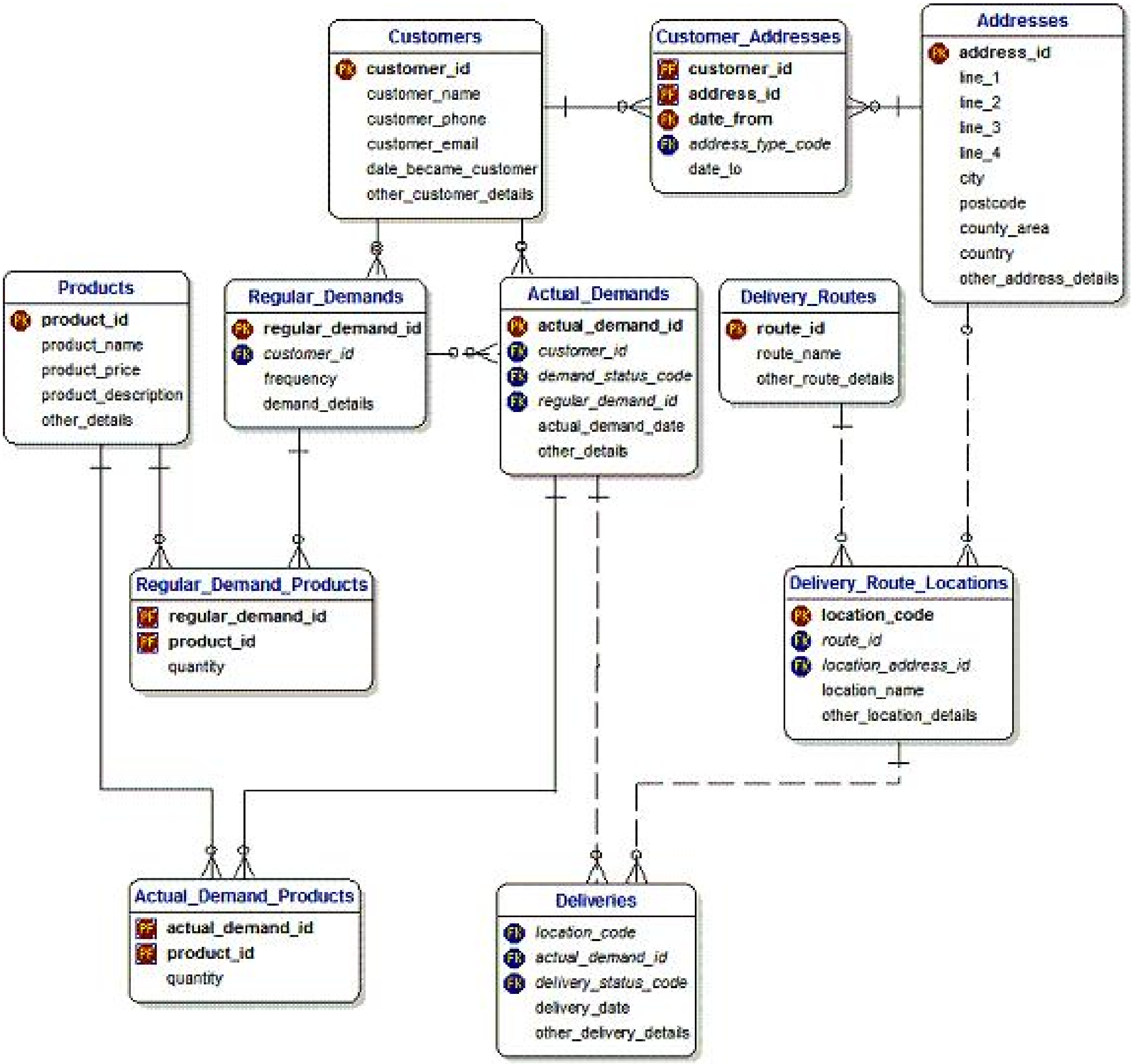
In this case, we could say :-

“A Customer can raise a Demand for Products to be delivered to a specified Address”



4.4.3 Deliveries – Complex

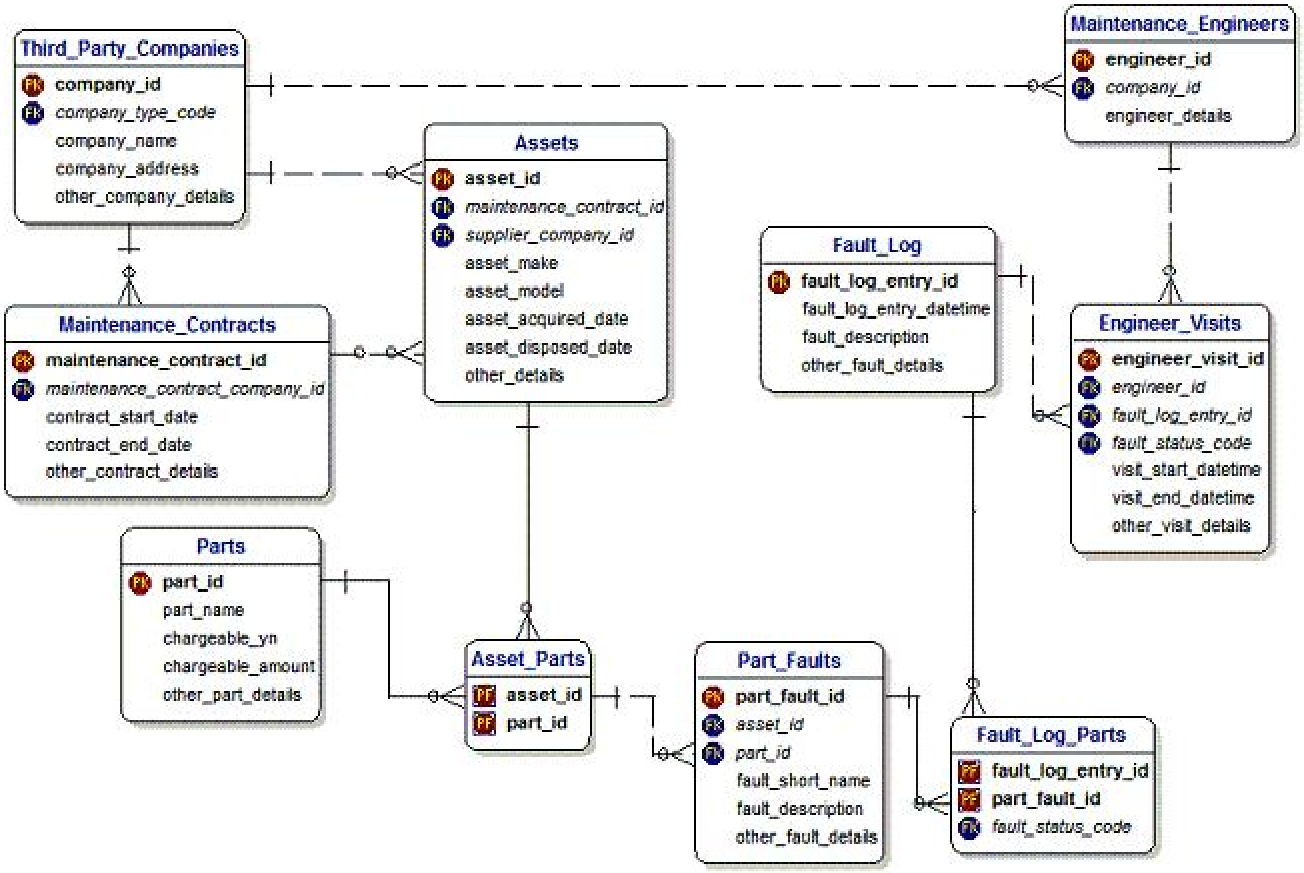
This shows a more complex Pattern which adds **Regular Demands.**



4.4.4 Maintenance of Equipment

The scope of this Data Model is the Maintenance of Assets by Third-Party Companies.
The Business Rules state :-

An Asset can have a Maintenance Contract.
An Asset consists of Asset Parts
Faults occur with these Parts from time to time.
Third Party Companies employ Maintenance Engineers to maintain these Assets.
Engineers pay Visits which are recorded in a Fault Log.
They correct the Faults and this is recorded in the Fault Log.



4.4.5 Top-Level and Subject Areas

Complex Data Models are common in large organisations.
They can best be understood when they are broken down into a Top-Level Model and Lower-Level Subject Areas.

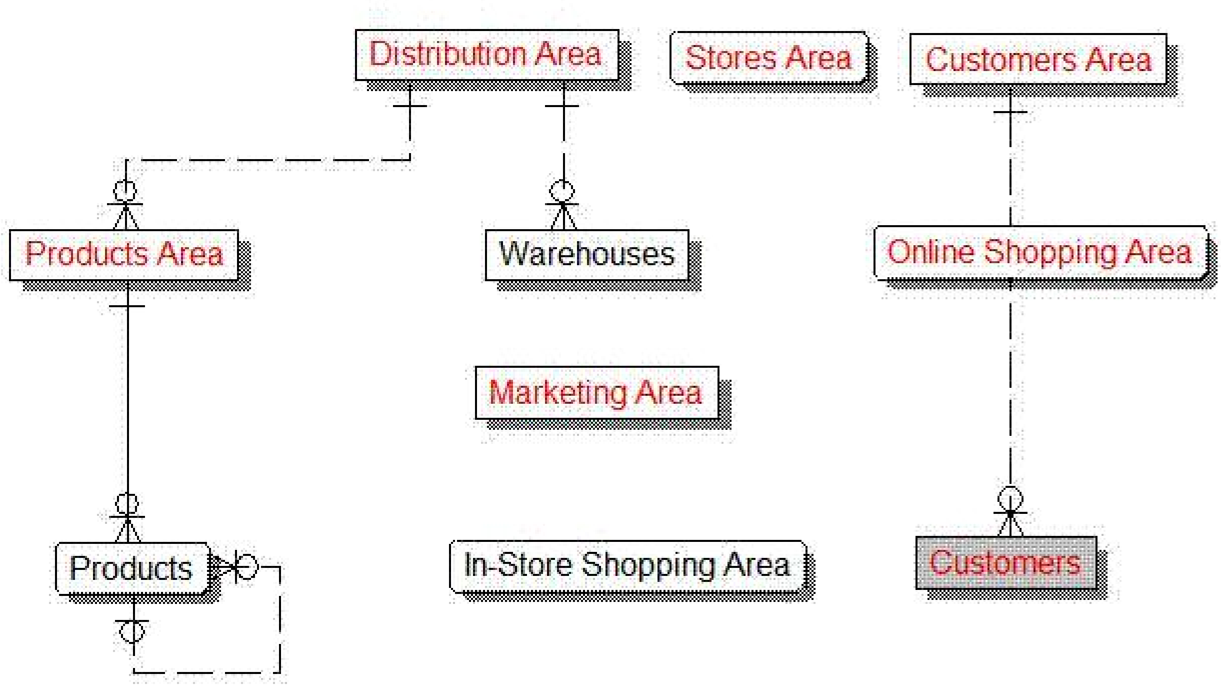
Typical Subject Area Models might include Finance, HR, Deliveries and Maintenance.

These are shown in earlier Sections of this document.

Top-Level Model
This is a top-level Model showing the Entities that are important at the top level.
It provides a suitable form of communication with a wide range of stakeholders.

A lower-level Model has been created for each specific Subject Area.

Here is an example for a Retail organisation that you can find on this page :-
http://www.databaseanswers.org/data_models/enterprise_data_model_for_retail/index.htm



4.6 What have we learned ?

This Chapter has discussed Design Patterns which are generic solutions to problems that occur frequently.

After we become familiar with these Design Patterns we can see them frequently when we look around us.

Therefore, an understanding of these Patterns can be a great help when we want to assemble a Data Model quickly and we can recognise that specific Patterns are relevant.