Database Answers

Learn Data Modelling

by Example

Barry Williams

Part 3 - Table of Contents

Welcome

This is the second Part of our three-part Book On Data Modelling.

This book has been produced in response to a number of requests from visitors to our Database Answers Web Site.
It incorporates a selection from our Library of about 950 data models that are featured on our Web site:

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

I hope you enjoy this Book and would be very pleased to have your comments at barryw@databaseanswers.org.

Barry Williams
Principal Consultant
Database Answers Ltd.
London, England

12. Check the Quality of a Data Model

12.1 Introduction

12.1.1 What is this?

This chapter discusses how to check the quality of a data model.

It builds through a series of structured steps.

These steps reflect the theory that underpins relational data model.

It concludes with a checklist for assessment of an overall quality.

12.1.2 Why is it important?

A data model often plays a fundamental part in the clarification of some key activities, such as the sources of data or the design of a database.

This makes it very valuable to be able to make an assessment of the quality of a data model.

12.1.3 What will I Learn?

You will learn a series of steps that follow a structured path to the formal assessment of whether a data model is fit for purpose.

12.2 Create a Top-Level Business Data Model

12.3.1 Types of Data Models

All the data models that we will be discussing can be described as Entity-Relationship Diagrams, or 'ERDs'. They all show relationships between entities or tables.

At the conceptual level, the 'things of interest,' such as 'customers,' are called *entities* and at the logical or physical level they are called *tables*, because they often appear as tables in databases.

At the physical level, tables are given names in the plural, such as *Customers*, whereas at the conceptual level they often appear in the singular, that is *Customer*.

At the logical level they might be either singular or plural.

A top-level business data model can be created using Microsoft Word and is intended for business users and a non-technical audience.

The other models referred to in this document will always be created by a data modeling tool such as ERWin or IBM's Rational Rose.

They could be described as conceptual, logical or physical models.

Conceptual models show the 'things of interest' that are in scope, for example, customers and materiel. They may or may not include keys and will certainly not include physical data types, such as the length of character strings.

Logical models will include primary and foreign keys and often the modeling tool will provide a facility to generate a physical model from a logical one.

Physical models are often close to the actual design of an operational database. They will always show data types and field lengths.

12.3.2 Example of a Simple Business Data Model

This model was created in Word and shows customers, orders and products. The flow of logic in a data model should go from top-left to bottom-right. This means that the more fundamental things are on the top and to the left.

This diagram is a good example:
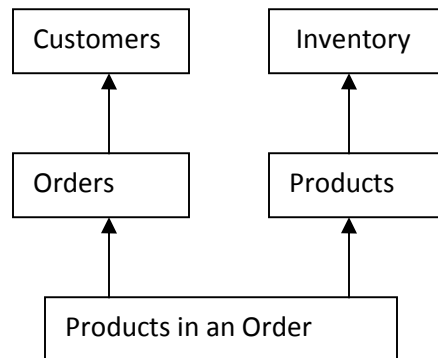
This version shows that customers and products each have a hierarchy so that a customer is part of a higher customer.

Similarly, a product can be part of a more complex product.

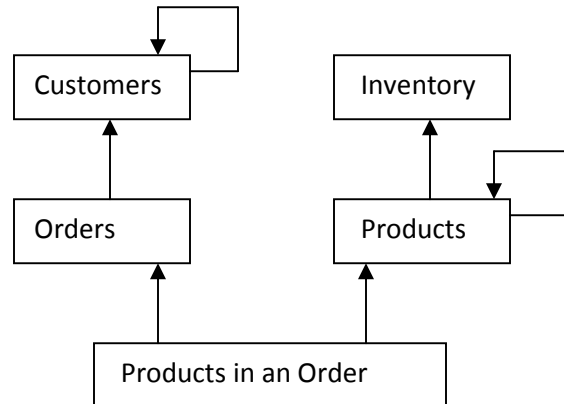Which of these two you choose to use will depend on the audience. In general, it is better to choose the simple option.

12.3 Draft the Business Rules

Business Rules are valuable because they define in plain English with business terminology the underlying relationships between the terms that appear in a data model.

The user 'commCustomery' will then be able to agree and sign off the rules.

Here is a small example:

Nr	TABLE	DESCRIPTION
D.1	Orders and Customers	An order must be raised by a valid customer. Not every customer will raise an order. Therefore the relationship must be one-to-many with a mandatory condition at the customer end and an optional condition at the order end.
D.2	Orders and Products	An order must refer to valid products. Therefore the relationship must be one-to-many with a mandatory condition at the product end and an optional condition

		at the order end, because not every product will appear in an order.
P.1	Products and Inventory	Products are kept in stores inventory

12.4 Draft a Glossary of Terms

It is very important to establish agreed definitions of terms and words in common use.

This is a small example:

TERM	DESCRIPTION	COMMENT
Order	A request for products to be supplied to the requesting customer.	
Product	A physical asset or service that can be separately ordered. It can be a component and a part of a larger assembly. It can be very small, such as a light bulb, or very large, such as a car	

12.5 Check that the Data Model is Correct

There may be errors that have a simple explanation. For example, the incorrect use of the modeling tool. Any errors should be discussed and resolved with the modeler and the users.

This is where the glossary and business rules are very valuable.

12.6 Review with Users

At this point, review the business rules and the glossary with users and aim to get sign off.

Make any necessary changes to format and contents.

12.7 Check Normalized Design

12.9.1 Normalized Design

This discussion applies to Entity-Relationship Diagrams (ERDs) and not to data warehouses.

We will start by defining the **rules for normalization** so that we can recognize cases where they have been broken.

Rules for Normalization

A little background is appropriate at this point.

The theory that provides the foundation for data models and ERDs was developed in 1970 by an Englishman called Ted Codd, who was a research scientist with IBM in California at the time.

Rule 1:

One of Codd's rules can be summarized as:

"The data in a table must belong to the key, the whole key and nothing but the key, so help me Codd".

This means, for example, that a record in a Customers Table must contain data only about the customer, and nothing about people in the customer, or activities of the customer.
It might include things like the name of the customer and when the customer was founded.

Check 1: Can the values of every data item in a table be derived only from the primary key?

Rule 2:

Another of Codd's rules stated that derived data must not be included.

For example, the headcount for a customer would not be included in the Customers Table because it can be derived by counting the records of members in the customer.

Check 2: Can any data item be derived from other items?

Rule 3:

There must be no repeating groups in a table.

The one uncomfortable exception is addresses. They are very often stored as a number of repeated lines called 'Address_Line_1,' 'Address_Line_2,' and so on.

Check 3: Do any column names repeat in the same table?

Rule 4:

An item of data must only be in one table.

For example, the name of a customer would appear only in the Customers Table.

Check 4: Does the same item of data in appear in more than one table?

12.8 Reference Data

12.10.1 Background
A list should be made of the reference data referred to in a data model.

When the list is complete it should be analyzed for consistency. For example, there will not usually be any relationships between the reference data. However, if there are any, then they should be sensible and consistent. For example, a town might be in a county which would be in a country. These could all be classified a reference data that has relationships that should be validated.

Typical reference data could include ranks, and types of materiel or equipment. In passing, we should note that customers, ranks and materiel are all examples of hierarchical structures. Ranks will change only very, very rarely. However, when they are stored in a table that is joined to itself then the table will have a recursive relationship to itself. Therefore, wherever these occur, we would expect to find compact data models that include a great deal with compact and powerful structures.

12.10.2 Standards
Any appropriate national, international standards must be considered when values for reference data are decided. These include MOD, NATO and ISO standards. For example, NATO maintains standards for product classification and this is already in use within the MOD. Therefore any data model relating to products should consider this standard and where appropriate the necessary tables should be added to the model.

12.9 Slowly Changing Data
The classic example of reference data that never changes is a calendar. The values are predictable for hundreds of years ahead. There is a category in between that is usually called **slowly changing data**. This applies where the values of the data changes on roughly a six-monthly basis.

Data about categories and types is often fixed values but some can change infrequently.

For example, a new aircraft type was introduced with unmanned aircraft. The values then became fixed-wing, rotary and unmanned. This would be an example of slowly changing data.

This highlights the fact that what constitutes reference data can be subjective and may be defined differently in data models created by different people or organizations.

12.10 Check Normalization

Check Nr	GOOD	OK (Y/N)?	DESCRIPTION
1	Y		Can the values of every data item in a table be derived only from the primary key?
2	N		Can any data item be derived from other data items?
3	N		Do any column names repeat in the same table?
4	N		Does the same item of data in appear in more than one table?

12.11 Look for Design Patterns

12.11.1 Some Examples

This data model shows examples of design patterns for one-to-many and many-to-many relationships, reflexive associations and reference data.

PK stands for 'Primary Key' and FK stands for 'Foreign Key'.

PF, which is shown in the Products_in_an_Order Table, stands for **P**rimary and **F**oreign key. This is a primary key in one table that is also a link to another table, where it is also a primary key.
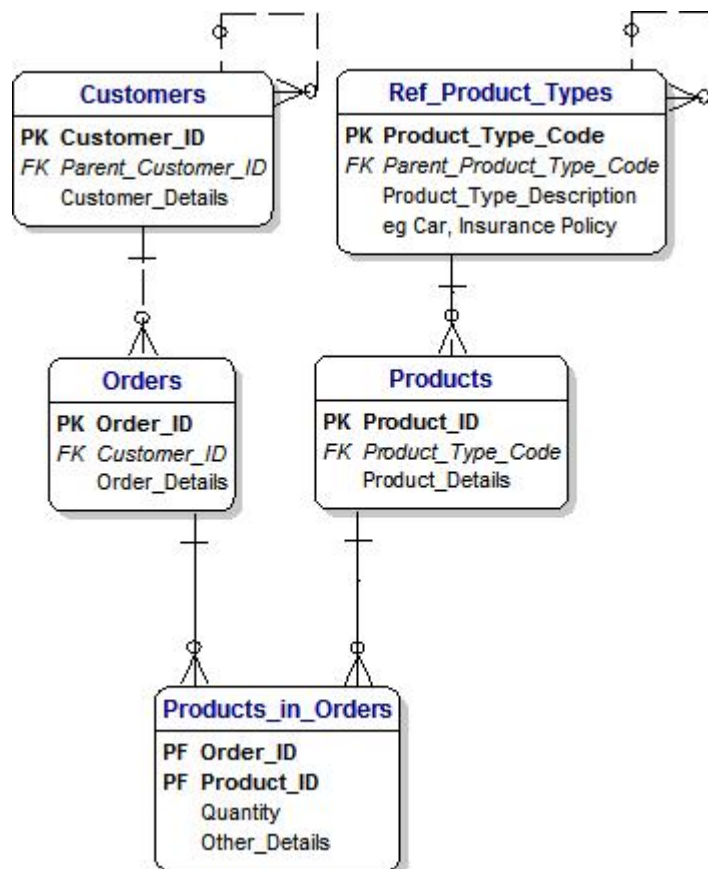
12.11.2 Inheritance

More details are provided in Chapter 3. Concepts in the document entitled "How to Understand a Data Model".

We use the concept of **inheritance** where have super-types and sub-types. Inheritance in data modeling is just the same as the general meaning of the word. It means that at a high level, we identify the general name of the 'thing of interest' and the characteristics that all of these things share. For example, an aircraft will have a name for the type of aircraft, such as *Tornado* and it will be of a certain type, such as fixed-wing or rotary.

At the lower level of fixed-wing aircraft, an aircraft will have a minimum length for the runway that the aircraft needs in order to take off.

This situation is shown in the following diagram:
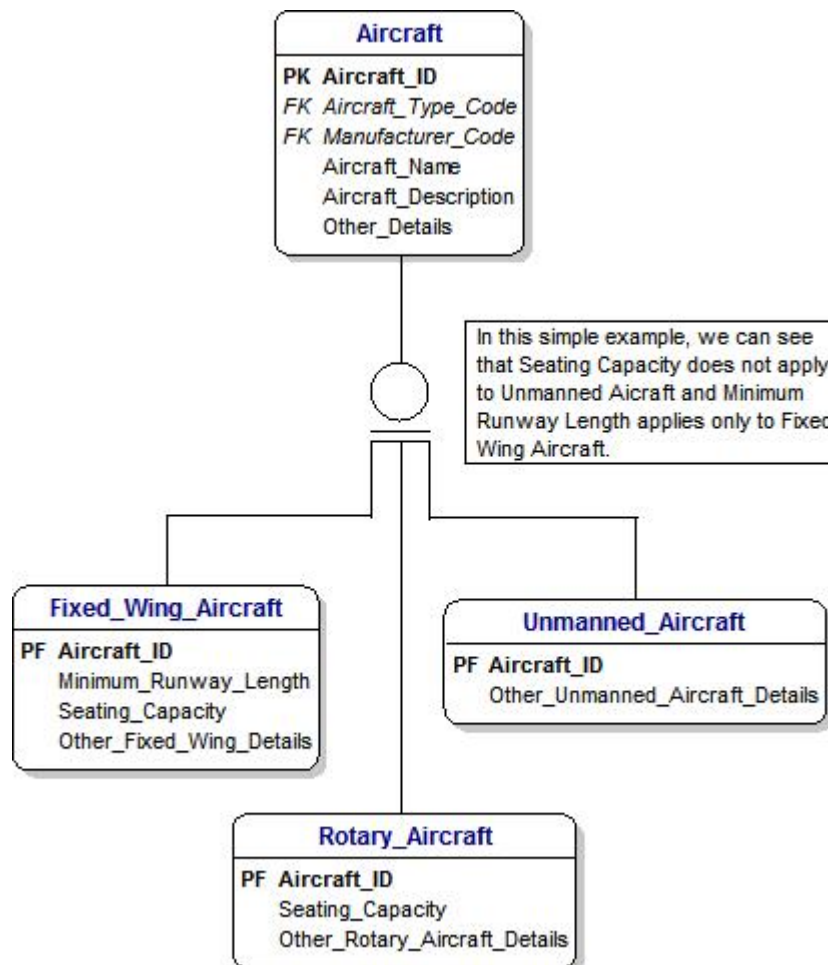
12.11.3 One-to-One Relationships

We can remind ourselves that Rule 1 above states:

"The data in a table must belong to the key, the whole key and nothing but the key, so help me Codd".

One implication is that there should not be a one-to-one relationship between two tables in a model because the data can be combined into one table with the same primary key. However, there is an exception to this which is when a one-off event can occur which involves a substantial amount of data. In that case, it would not be good to create a large number of fields which will be blank in the large majority of cases.

For example, when a soldier joins the army there might be data that is involved only with the joining details. The basic data for the soldier will be part of his or her basic records – such as date of birth and place of birth. If a separate table exists for 'Joining Details' then it would contain such things as date and place of joining. Then the Soldiers Table would have a one-to-one relationship with the Joining Details Table.

In other words, it can sometimes be acceptable to see a one-to-one in a data model. If that happens, it is necessary to establish the associated business rules to clarify the conditions.

12.12 Review Data Warehouses Designs

This section is relevant if the data model includes a data warehouse or data mart. A data warehouse can be a star or a snowflake design.

This diagram shows a typical data warehouse. It is a star structure with only one dimension for the related dimension tables. The arrows point from children to parents. This is a simple data warehouse for customers, orders and products.

Calendar

Customers

Locations

Facts

Dimensions -

Date

Customer ID

Job Title

Location ID

Order ID

Product ID

Product Type Code

Facts of Data -

Date of Demand

Products in Demands

Unit raising Demand

Averages, Counts, Totals

KPIs and Dashboards

Other Derived Figures

Job Titles

Orders

Products

Product Types
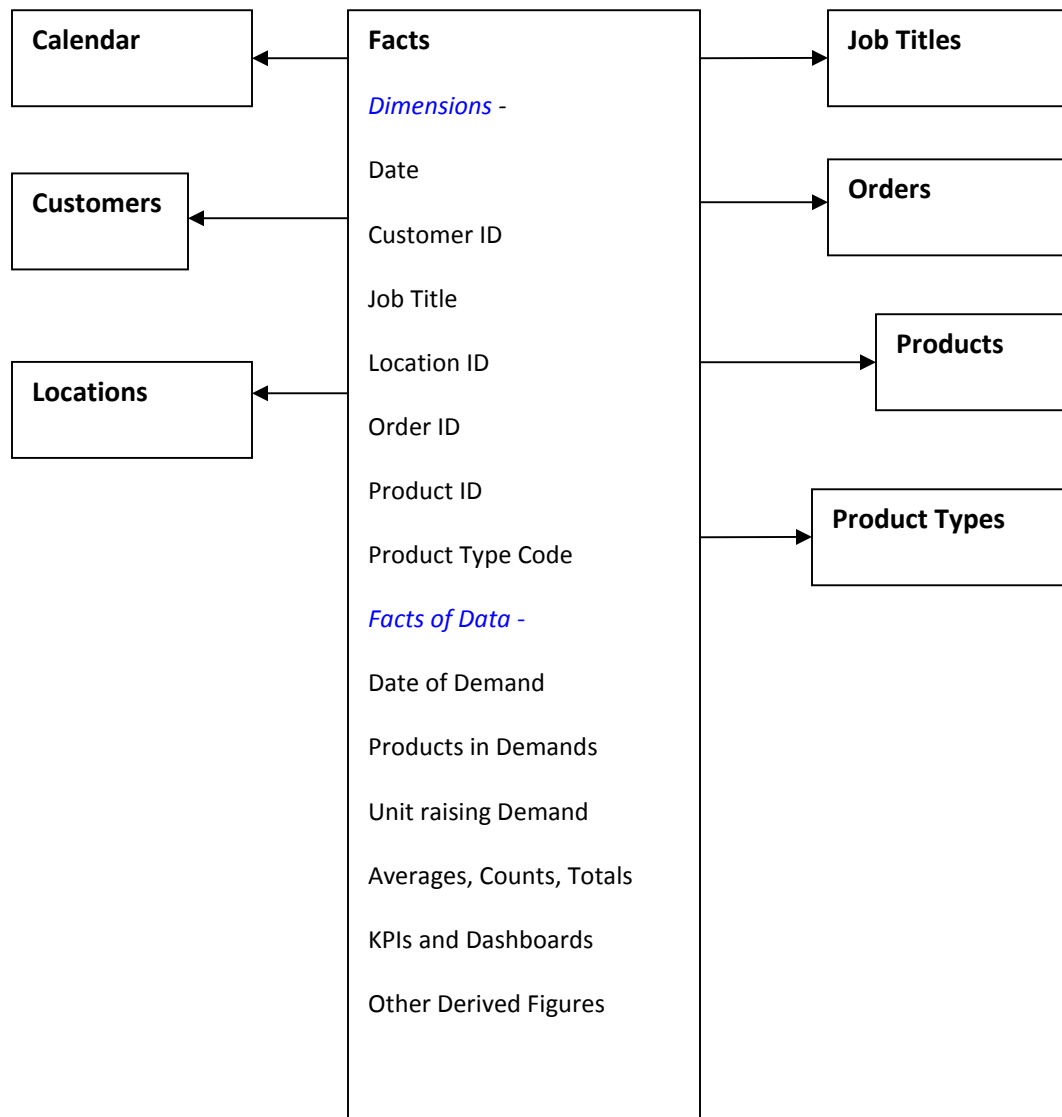
12.13 Check Naming Standards

At this step, we check for compliance with naming standards. For example, a typical standard might state that field names should be specified with underscores linking related words and first letters in capitals, such as Customer_ID. In the absence of any explicit standard, this should be the default.

This is shown in the model in Section 9.1 and also in this one.

We might say that naming standards are nice to have. In other words, they are not essential but they reflect best practice.
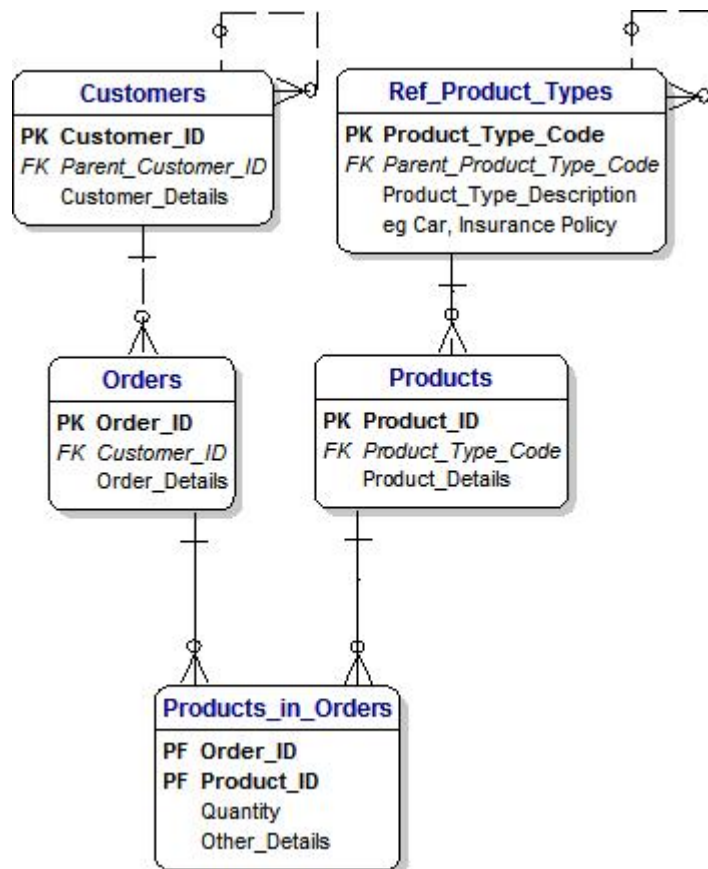
12.14 Check for Consistent Data Types

There are two reasons why it is important to check for consistent data types and lengths:

1. It avoids nasty surprises when a physical database is generated from the data model.

2. It is an indication of the professionalism of the manner in which the data model was produced, unless it has been reverse engineered from a database, in which case these design considerations do not apply.

For example, names should always be the same, or should be handled in a way that handles any differences in a way that ensures consistency.

Typically, a longer name should be explicitly truncated to a shorter value where appropriate.

In the absence of any explicit standard, the default for names or address lines should be VARCHAR(255) or VARCHAR2(255) for Oracle.

For other character strings they should default to *Memo* or *Text*.

12.15 Check for Defaults

We would like to see Default Values used wherever possible because they increase the discipline enforced by the model and they indicate that a thorough analysis was carried out during the creation of the data model.

For example, a 'Start Date' could default to the current day, or the 'System Date'.

12.16 Determine the Assurance Level

The **assurance level** could be:

i. Acceptable
ii. Acceptable with reservations
iii. Not acceptable

12.17 What have we learned?

In this chapter we have learned a structured approach to checking the quality of a data model produced by somebody else.

We have learned to determine whether it is fit for purpose.

Our approach is based on the sound theoretical foundation that is a very strong part of data models for relational databases.

13. Enterprise Data Models

13.1 Introduction

This chapter will discuss **enterprise data models** for a number of different kinds of enterprise.

13.1.1 What is this?

This chapter is a review of techniques suitable for designing large-scale data models.

13.1.2 Why is it important?

Being able to create enterprise data models will help you to handle the most challenging situations that you might encounter as a data modeler.

The Best Practice Approach has these steps:

- Create a top-level data model that represents a business-eye view of the organization.

  - It will include all the subject areas, such as marketing, operations, finance and CRM.

  - Each of these subject areas will have its own data model.

13.1.3 What Will I Learn?

Enterprise data models are very useful because they help to establish 'A single version of the truth'.

For example, it is common for organizations to have multiple occurrences of customer records, with the same person appearing as Joe Bloggs, Joseph Bloggs, Joey Bloggs and so on. These can be matched to one common record and all the associated details can be consolidated. This aspect is discussed in more detail in Chapter 10, which covers Master Data Management.

The approach we adopt is to define a **statement of objectives** that will establish the important 'things of interest' that need to be in the enterprise data model. This establishes a context from which subject areas can be derived. Any change in the statement of objectives can easily be reflected in changes to the subject areas.

13.2 Business Intelligence (BI)

13.2.1 Statement of Objectives

Our example is an Anglo-Swiss Consulting Company, which has offices in London and Zurich. The typical statement of objectives for this company would be stated in these terms:

"To provide high-level consulting services to a targeted clients operating from multiple locations."

To support these Objectives, we have designed a BI approach called BMEWS, which stands for 'Business Monitoring and Early Warning System'.

"To provide a flexible solution to a range of user requirements for Business Intelligence."

Therefore, the subject areas in the enterprise data model must include:

Consulting:

- Clients

- Costs

- Revenues

- Services

- Staff

BMEWS:

- Data warehouse and data marts

- Key Performance Indicators (KPIs)

- Data sources and integration

13.2.2 The Enterprise Data Model

This model appears on this page on the Database Answers Web site:

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

Enterprise Data Model for BMEWS BI
Barry Williams
Principal Consultant
DatabaseAnswers.com
April 19th 2011
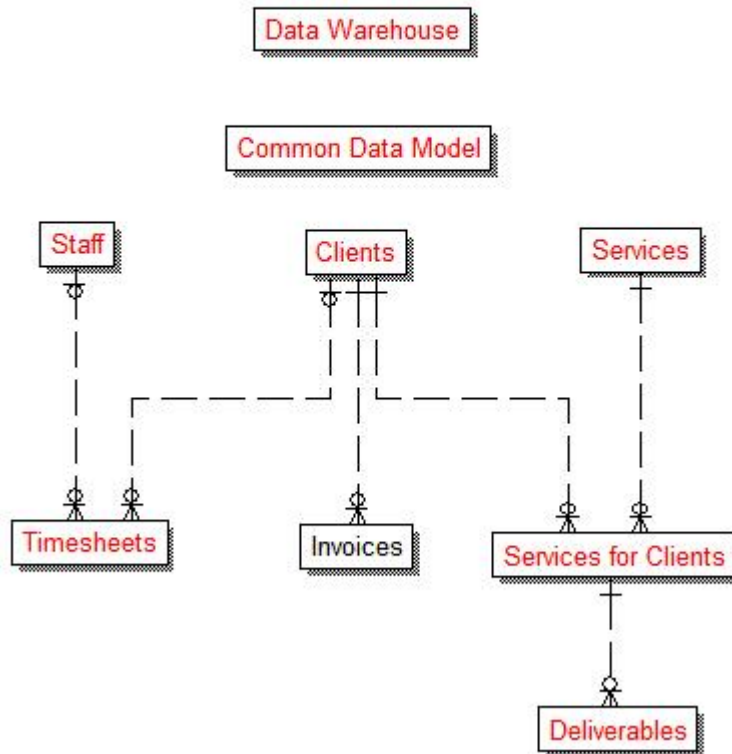
13.2.3 BI Data Warehouse

This model appears on this page on the Database Answers Web site:

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

Data Warehouse for BMEWS BI
Barry Williams
DatabaseAnswers.org
April 19th. 2011
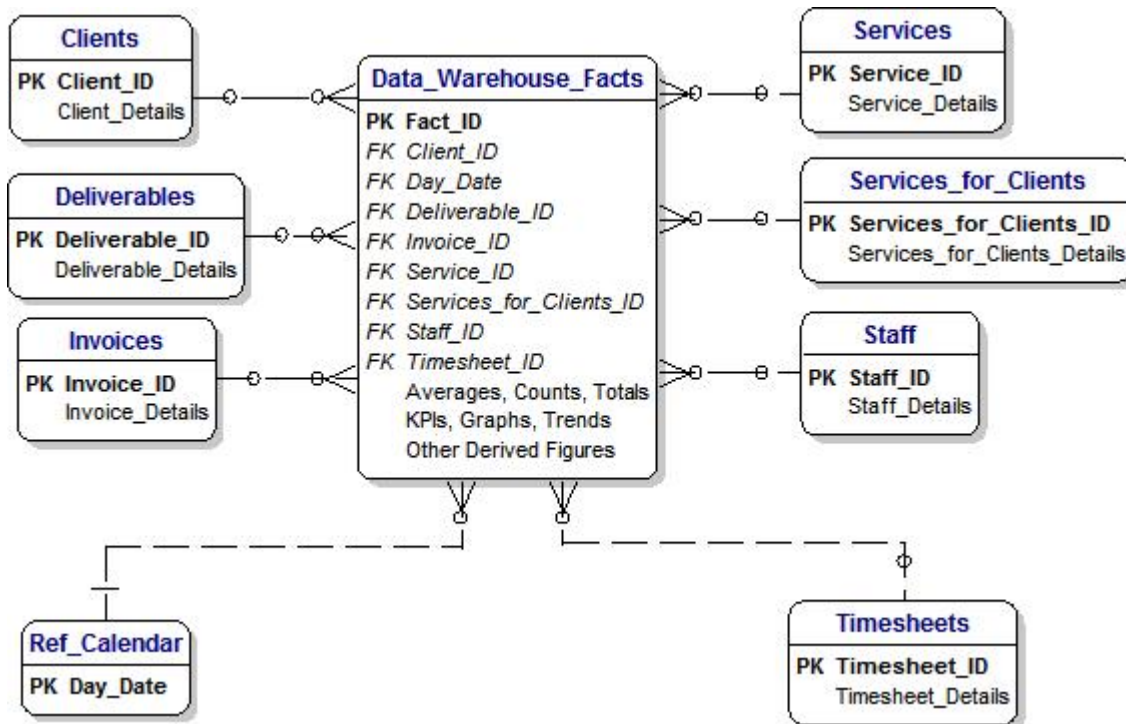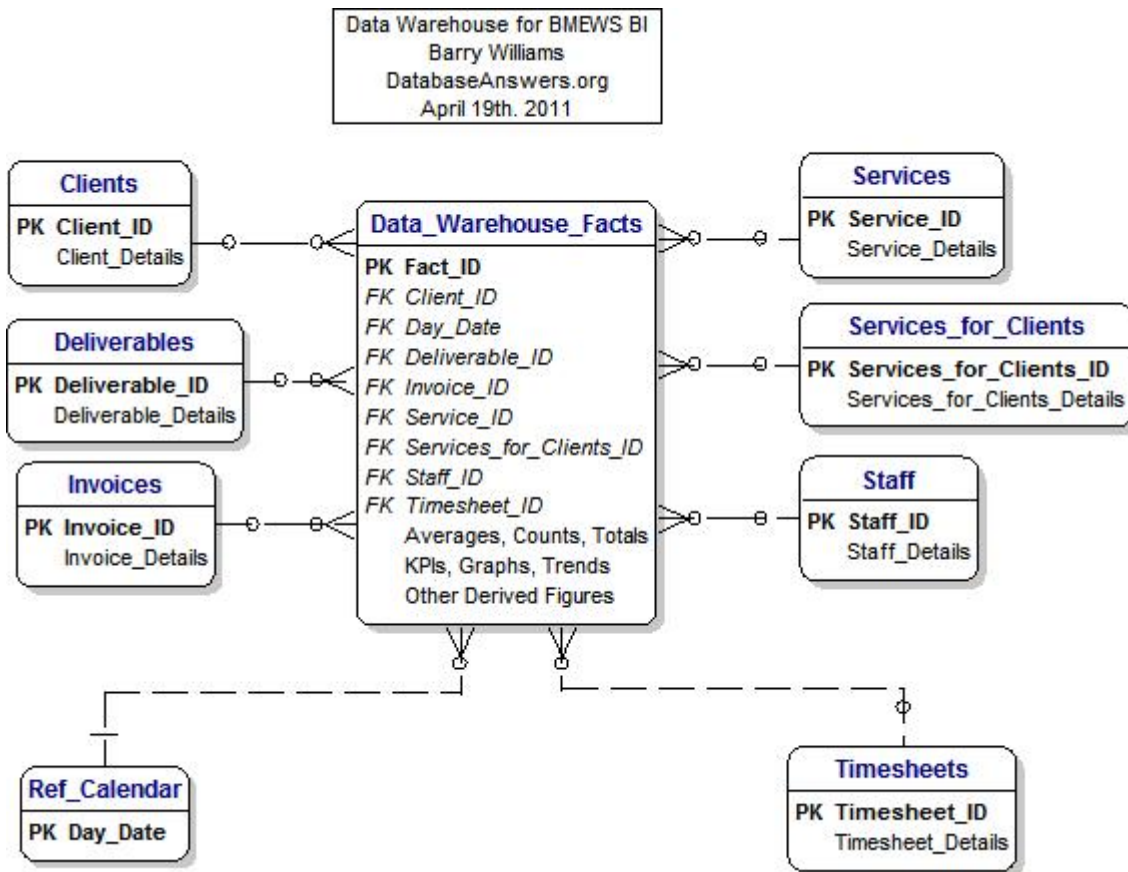
13.3 Insurance

13.3.1 Statement of Objectives

The typical statement of objectives for an insurance company would be stated in these terms:

"To provide a wide range of attractively-priced and economical services to a targeted range of customers, with economic management of claims processing and settlements."

Therefore, the subject areas in the enterprise data model must include:

- Claims

- Customers

- Policies

- Products and services

- Staff

- Settlements

13.3.2 The Enterprise Data Model

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/enterprise_data_model_for_insurance/index.htm
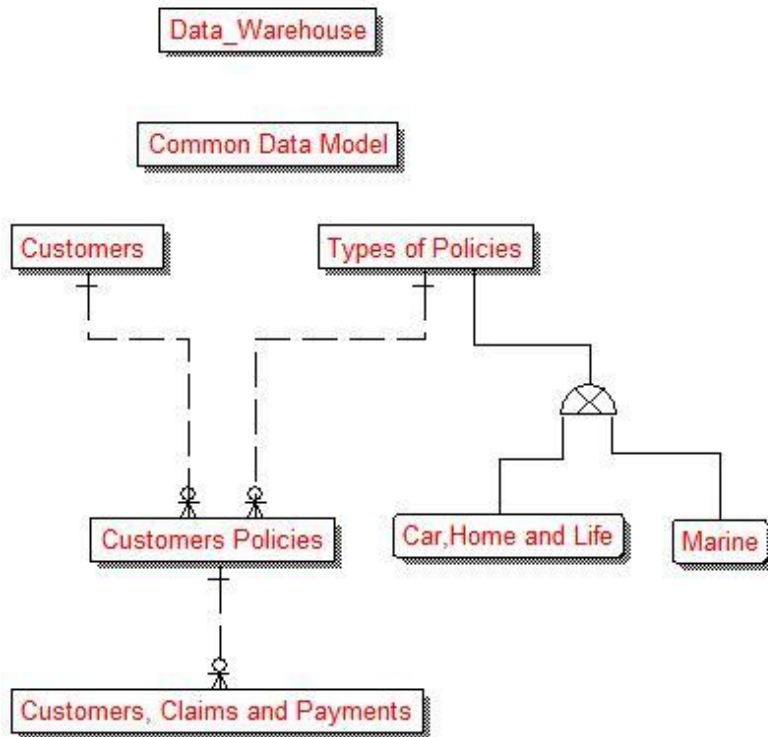
13.3.3 The Customers, Claims and Brokers Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/insurance_customers_and_claims/insurance_brokers_erwin_physical.htm
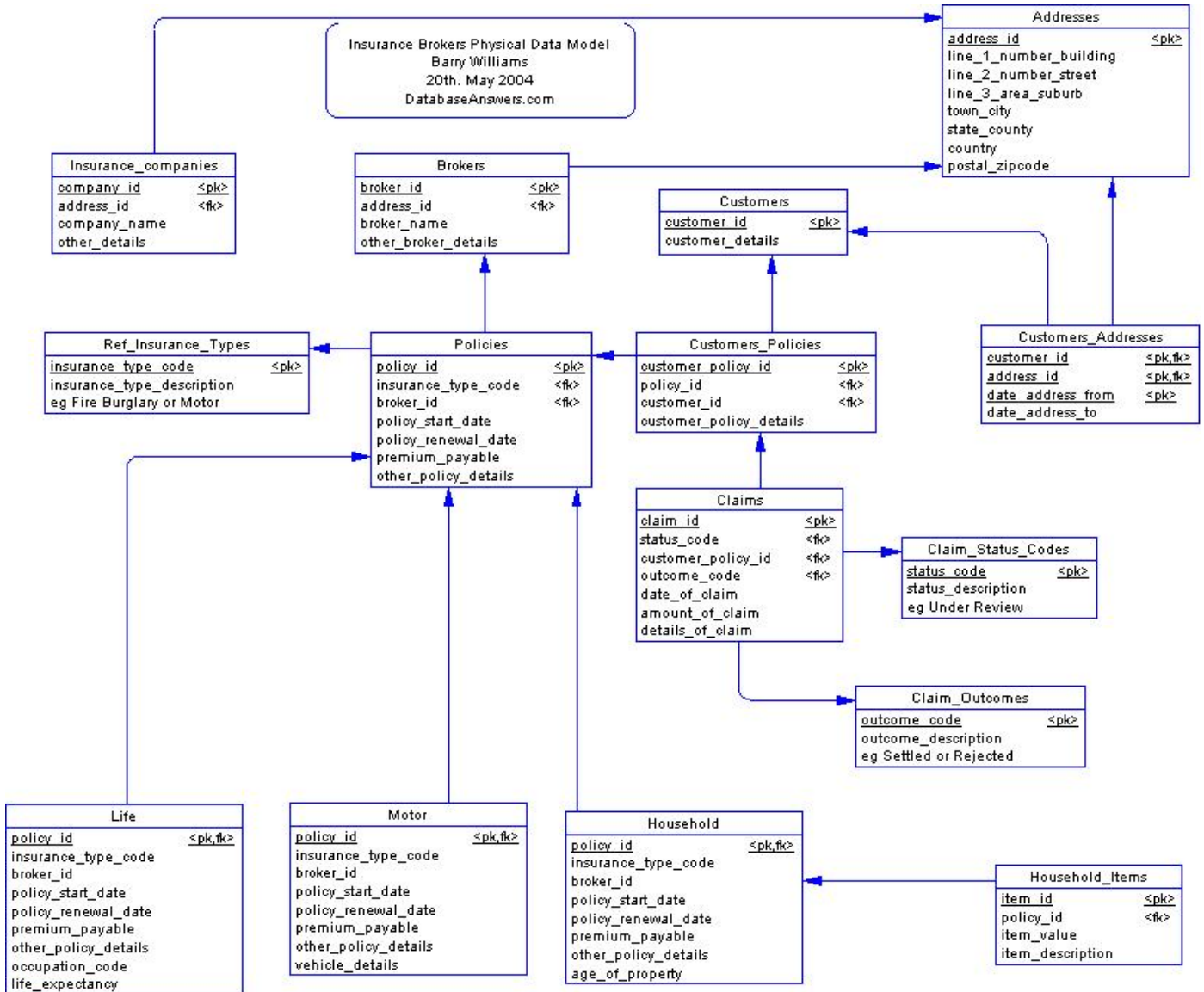
13.3.4 Policies and Claims Subject Area

This model appears on this page on the Database Answers Web site:

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

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

Data Model for Insurance Claims
Barry Williams
DatabaseAnswers.org
April 14th. 2011

13.3.5 Types of Insurance - Car, Home and Life Subject Area

This model appears on this page on the Database Answers Web site:

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

Policy_Types

PK Policy_Type_ID
FK *Policy_Type_Code*
   Other_Details

Ref_Policy_Types

PK Policy_Type_Code
   Policy_Type_Name
   eg Car, Home, Life

Ref_Car_Policy_Types

PK Car_Policy_Type_Code
   Car_Policy_Type_Name
   eg Comprehensive
   eg 3rd Paty, Fire, Theft

Car_Policies

PF Car_Policy_Type_ID
FK *Car_Policy_Type_Code*
   Car_Manufacturer_Code
   Car_Model_Code
   Other_Details

Home_Policies

PF Home_Policy_Type_ID
FK *Address_ID*
   Date_House_Built
   Other_Details

Life_Policies

PF Life_Policy_Type_ID
   Medical_History
   Prescription_Details
   Other_Details

Addresses

PK Address_ID
   Address_Details

13.3.6 Types of Insurance – Marine and Motor Vehicles Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/insurance_general/index.htm
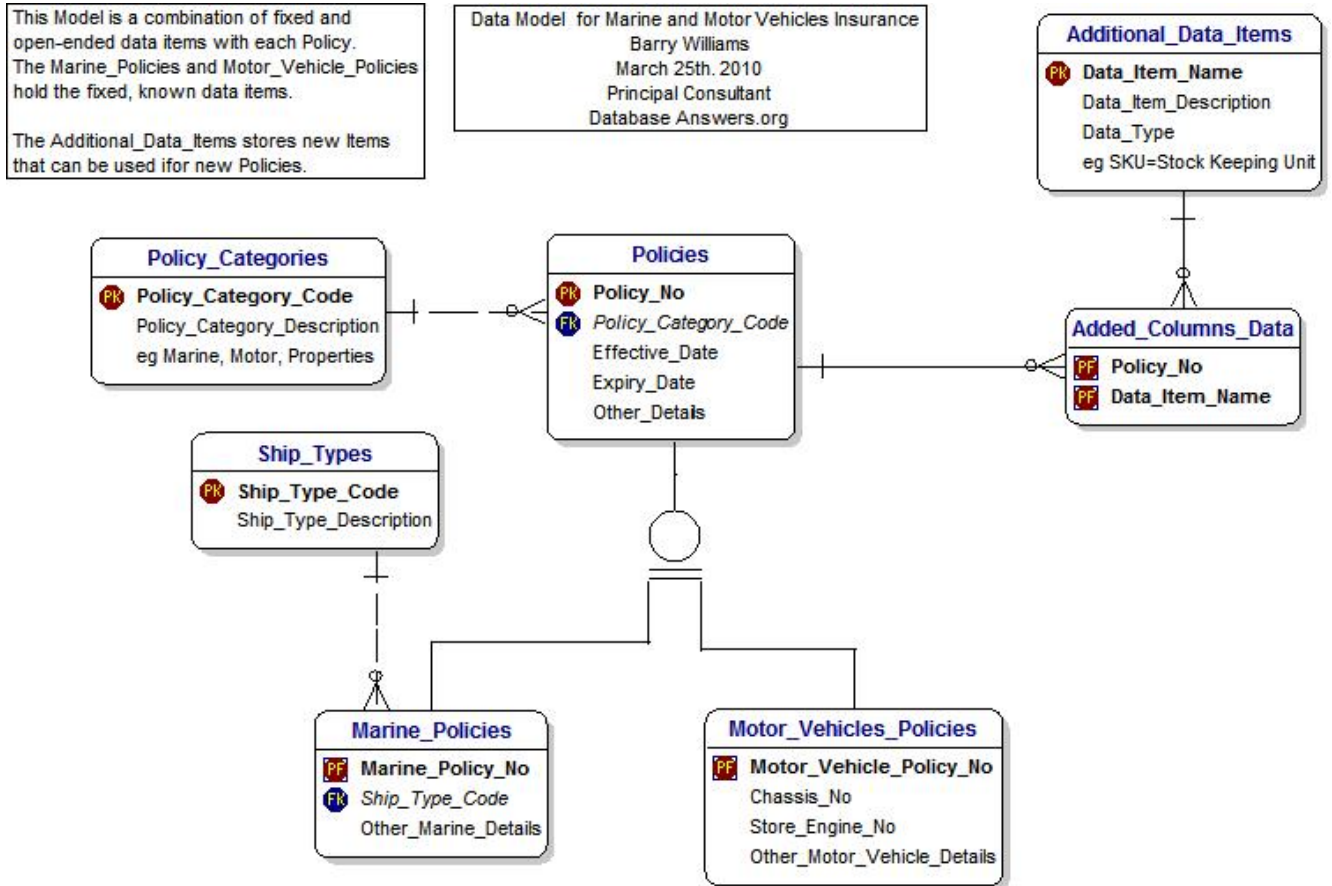
13.3.7 Common Data Model

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/insurance_data_warehouses/common_data_model.htm
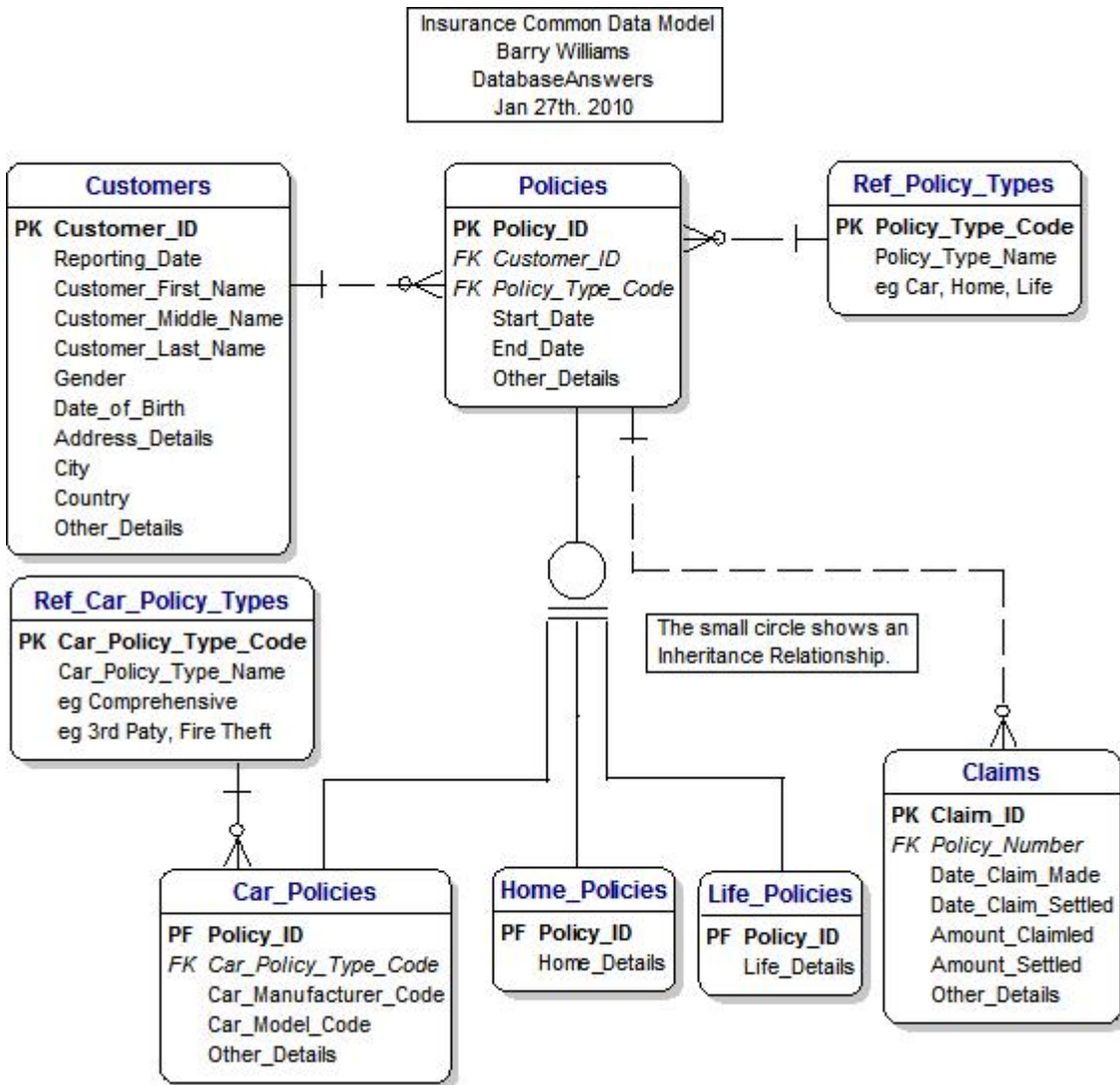
13.3.8 Insurance Data Warehouse

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/insurance_data_warehouses/index.htm
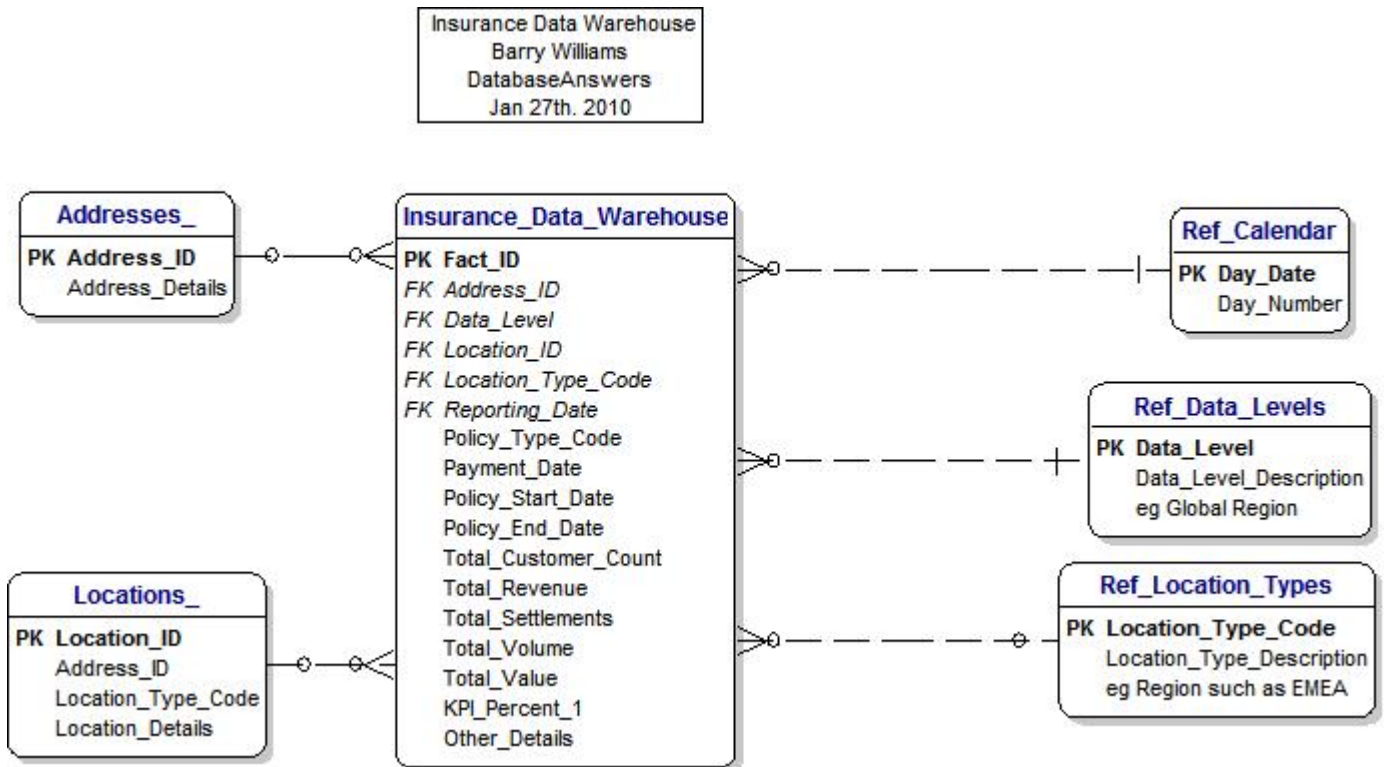
13.4 Investment Banking

13.6.1 Statement of Objectives

The typical statement of objectives for an investment bank would be stated in these terms:

"To participate in a profitable trading operation by buying and selling stocks, shares and other commodities."

Therefore, the subject areas in the enterprise data model must include:

- Accounts

- Clients

- Financial products and services

- Staff

- Settlements

- Trades or deals

13.6.2 Enterprise Data Model

This model appears on this page on the Database Answers Web site:

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

Enterprise Data Model for Investment
Banking
Barry Williams
Principal Consultant
DatabaseAnswers.com
March 26th 2011
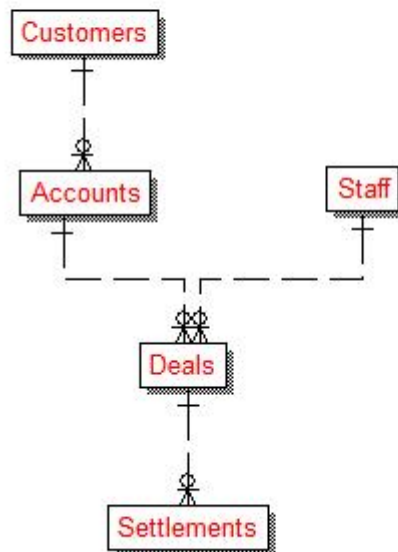
13.6.3 Accounts Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/investment_banking/accounts.htm

Investment Banking Accounts Model
Barry Williams
Principal Consultant
DatabaseAnswers.com
23rd. July 2004

13.6.4 Customers Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/investment_banking/customers.htm

Investment Banking - Customers
Barry Williams
DatabaseAnswers.com
25th. July 2004

13.6.5 Deals Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/investment_banking/deals_general.htm

13.6.6 Settlements Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/investment_banking/settlements.ht m

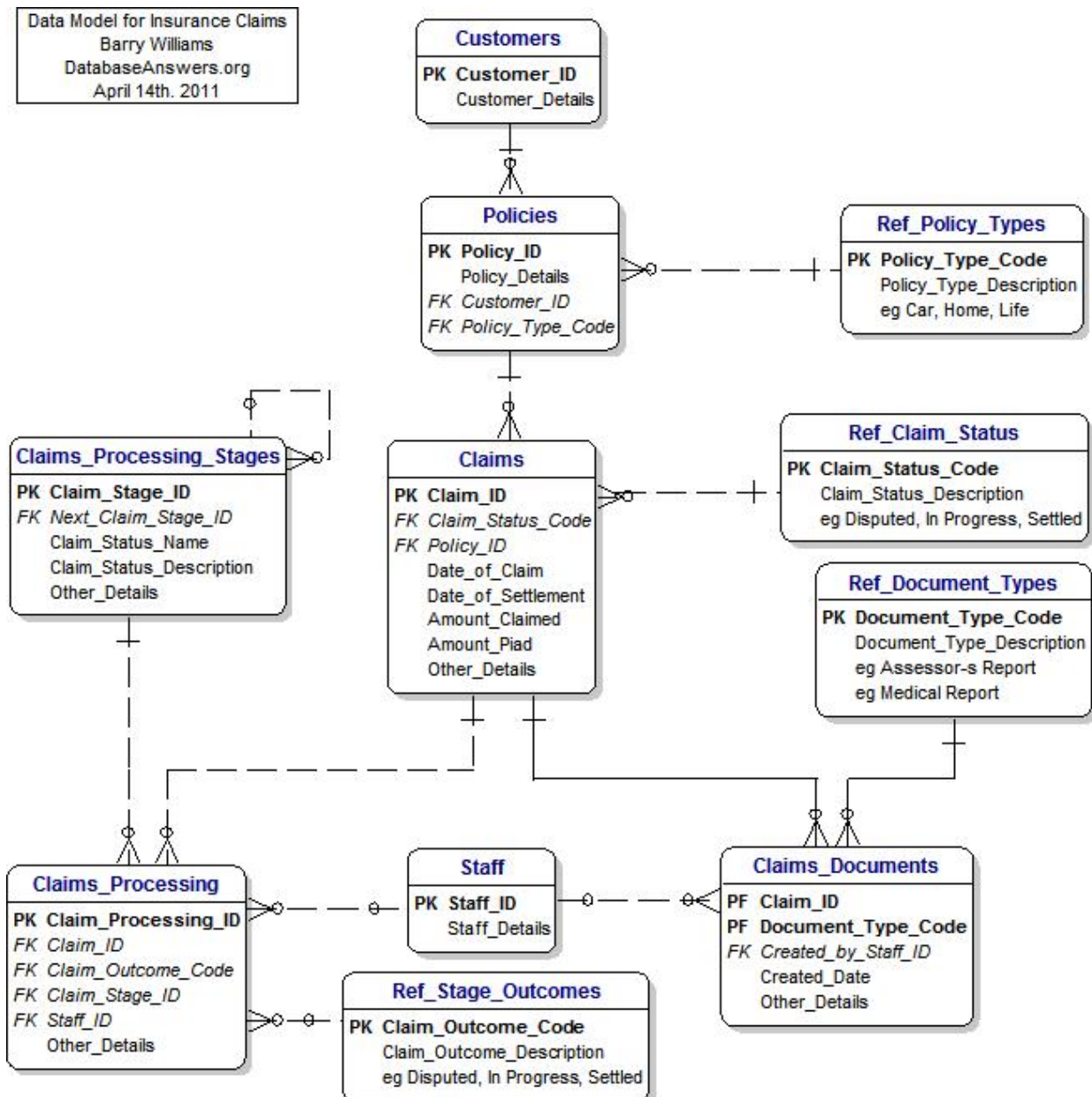Investment Banking - Settlements
Barry Williams
DatabaseAnswers.com
25th. July 2004
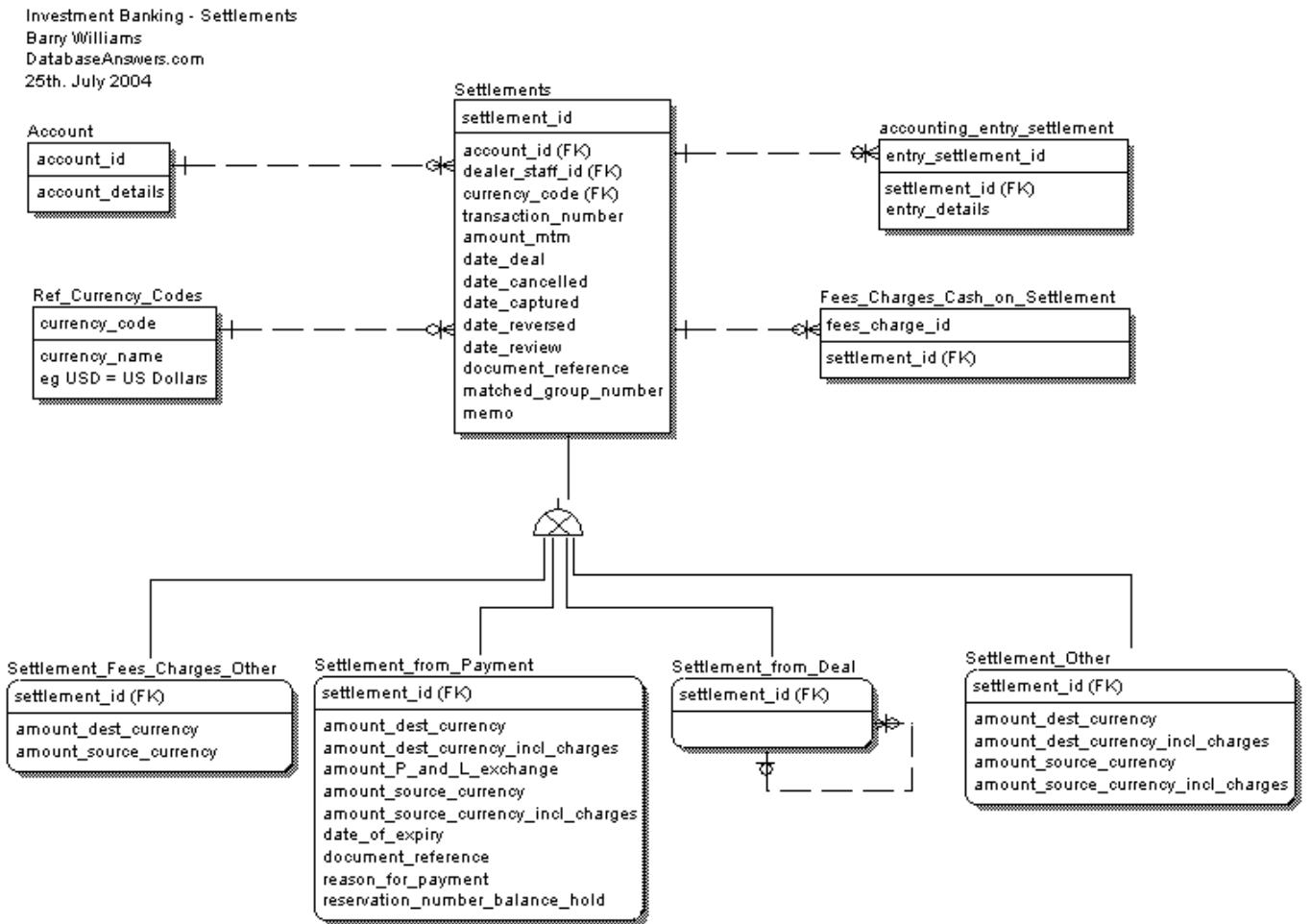
13.6.7 Staff Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/investment_banking/staff.htm

13.6.8 Data Warehouse

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/banking_data_warehouses/investment_banking_data_warehouse_final_version.htm
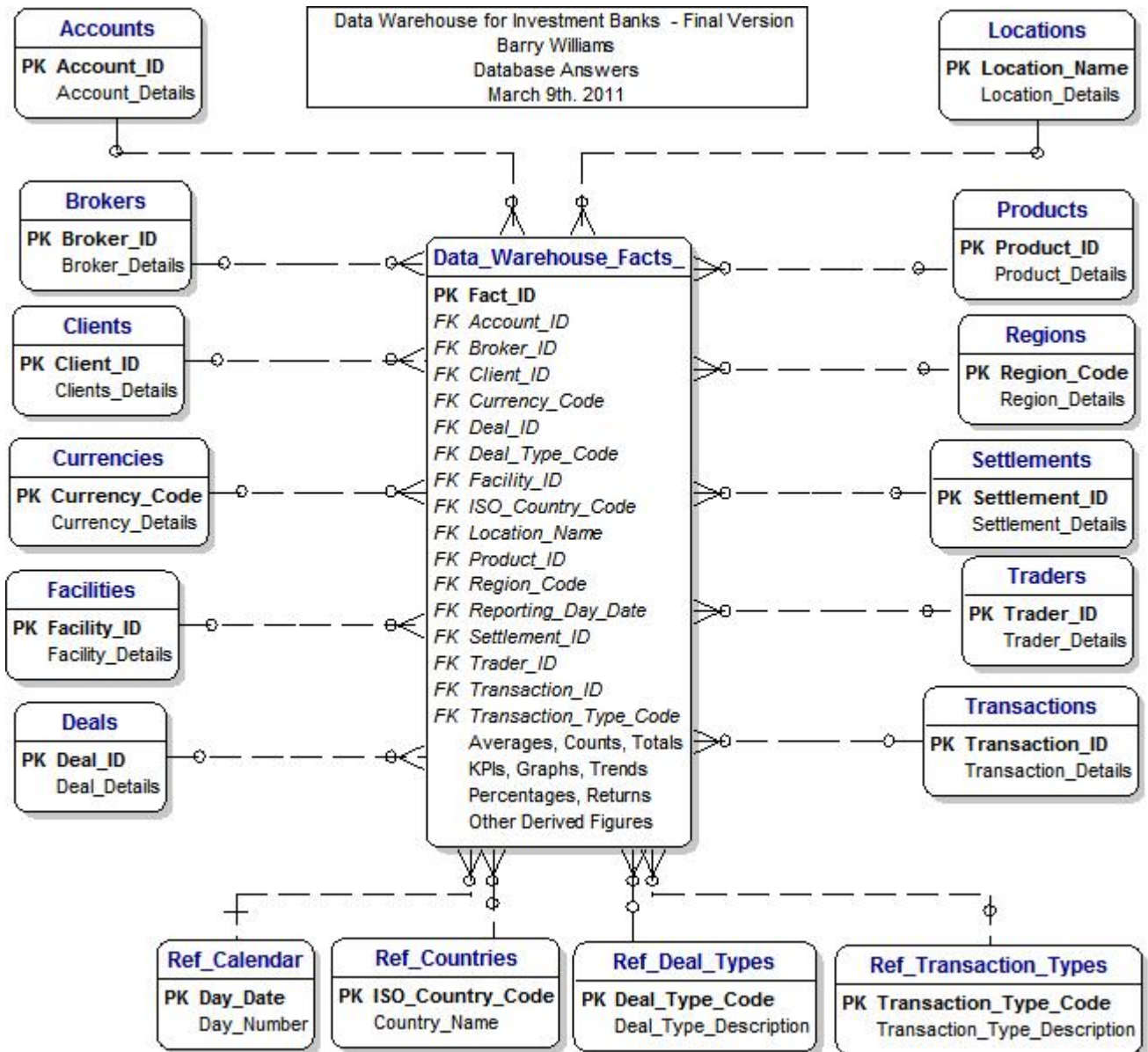
13.5 Local Government

13.7.1 Statement of Objectives

The typical statement of objectives for a UK local government would be stated in these terms:

"To provide a range of services to a varied range of non-residents and residents. To ensure that these services are provided economically to meet the specific requirements of the community."

Therefore, the subject areas in the enterprise data model must include:

- Customers or citizens, with both residents and non-residents.

- Services and products

- Service delivery

13.7.2 Enterprise Data Model

This model appears on this page on the Database Answers Web site:

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

Enterprise Data Model for Local Gov't
Barry Williams
Principal Consultant
DatabaseAnswers.com
March 26th 2011
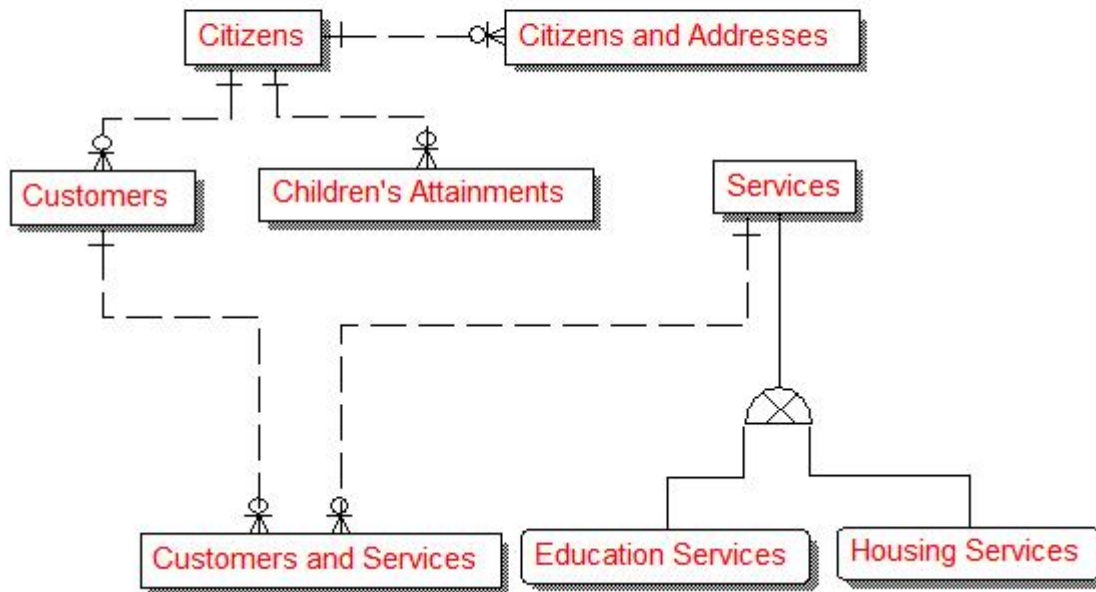
13.7.3 Children's Attainments Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/enterprise_data_model_for_local_government/children.htm
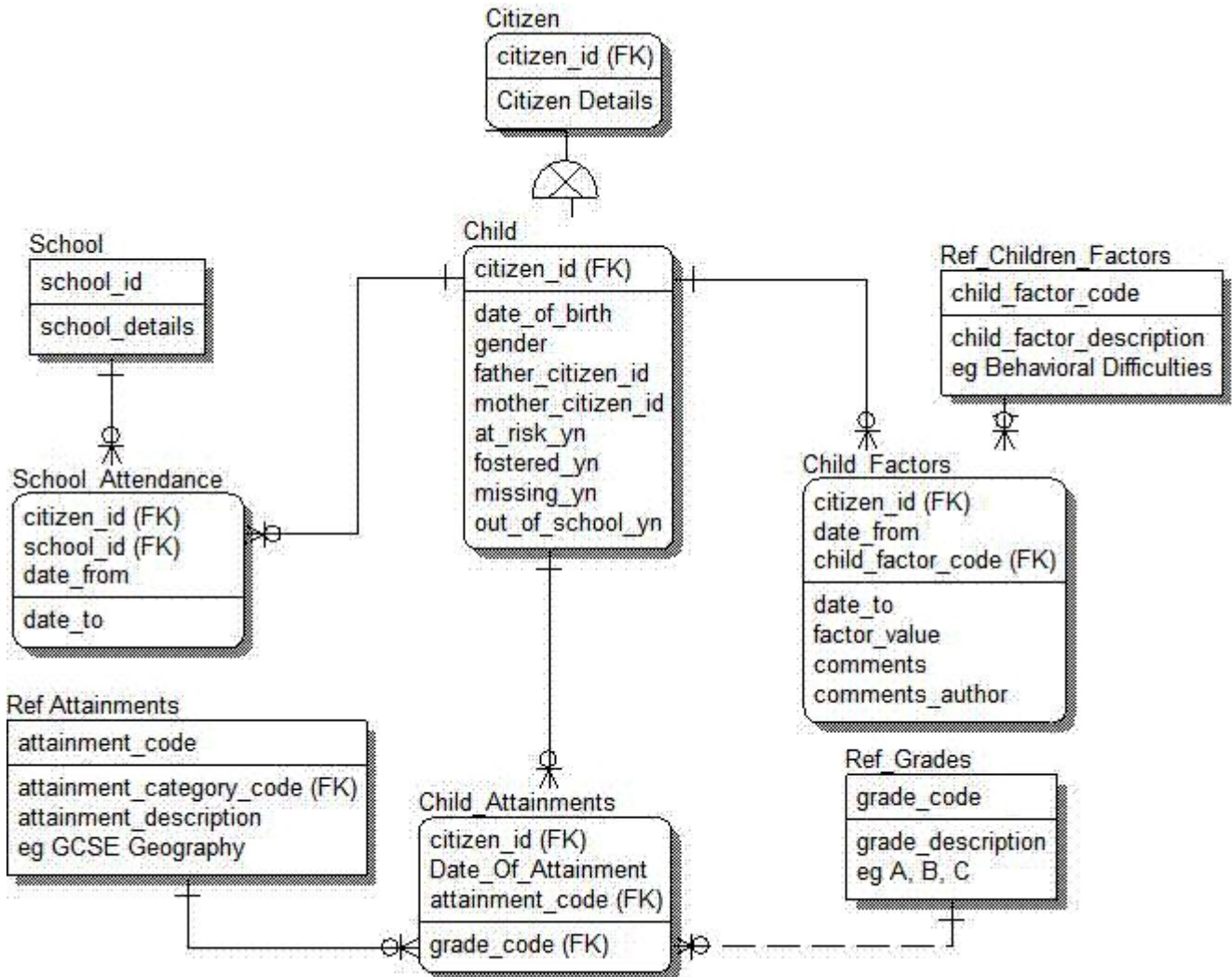
13.7.4 Citizens Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/enterprise_data_model_for_local_government/citizens.htm
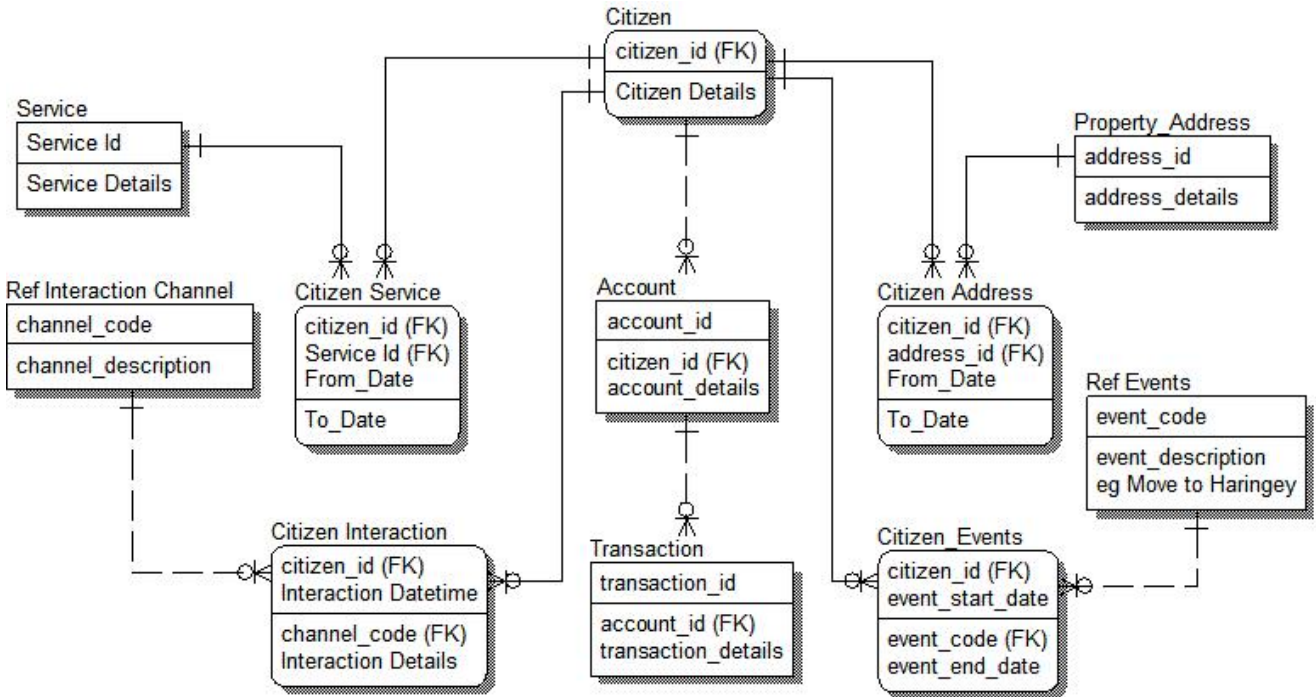
13.7.5 Citizens and Addresses Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/enterprise_data_model_for_local_government/citizens_and_addresses.htm

Citizens and Addresses Data Model
10.00 am Friday
29th. October 2004
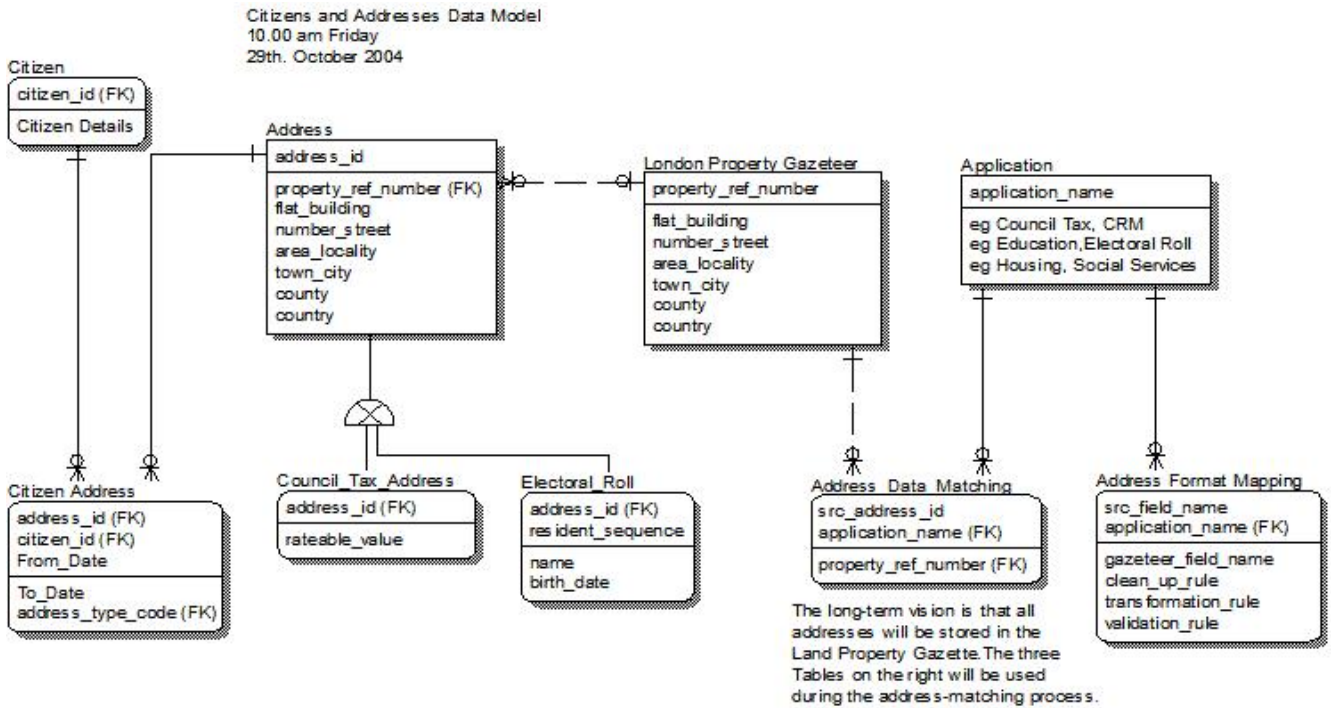
13.7.6 Citizens and Services Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/enterprise_data_model_for_local_go vernment/customers_and_services.htm
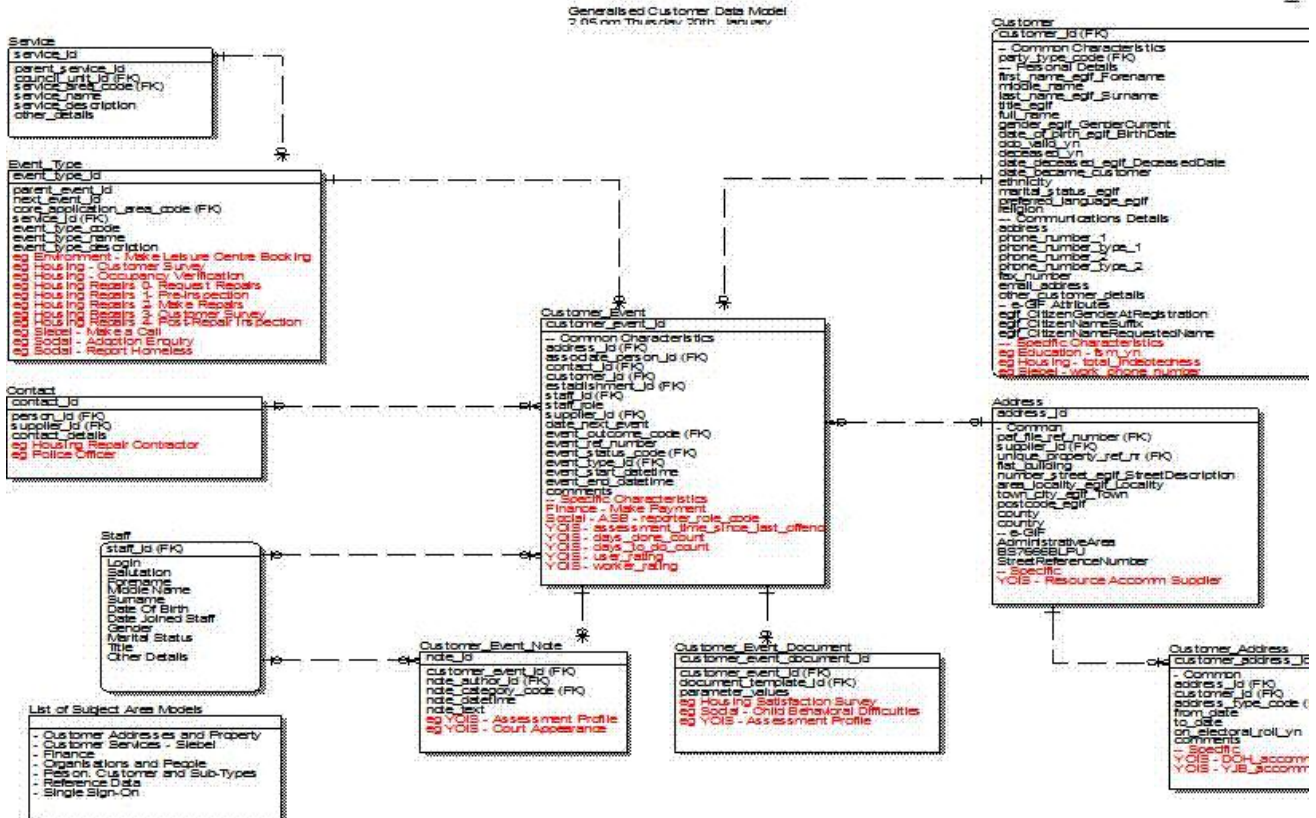
13.7.7 Education Services Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/enterprise_data_model_for_local_go vernment/education_services.htm

Person
- person_id
- address_id
- surname
- forename
- title
- initials
- birthdate
- gender
- deceased_ind
- notes_ind
- other_details

Person Address
- person_id (FK)
- address_id (FK)
- address_start_date
- address_end_date

Establishment
- DFEE_no
- address_id (FK)
- estab_id
- LEA_no
- estab_name
- notes_ind
- other_details

FSM Eligible
- pupil_id (FK)
- accurate_at
- fsm_ind

Pupil Guardian
- pupil_id (FK)
- person_id (FK)
- relationship
- responsibility

Address
- address_id
- UPRN
- house_no
- unit_no
- unit_name
- street
- main_road
- district
- town
- county
- postcode
- country

User Def Groups
- subject
- group_code
- group_code_desc

Round Pupil
- pupil_id (FK)
- round_code
- APP_REF

Pupil
- pupil_id
- FORMER_UPN
- NATIONAL_PUPIL_ID
- NHS_NUMBER
- cohort
- surname
- forenames
- known_name
- birthdate
- gender
- notes_ind
- deceased_ind
- current_pupil
- other_details

Pupil Prof
- pupil_id (FK)
- person_id (FK)
- person_type
- module
- agency_id
- not_known
- correspondence

Info
- info_id
- module
- info_text

User Def Value Code
- subject (FK)
- group_code (FK)
- value_code
- value_code_desc
- data_type
- max_length

SEN Level
- pupil_id (FK)
- sen_level
- accurate_at
- sen_type
- referral_id
- statement_id

Pupil Address
- pupil_id (FK)
- address_id (FK)
- address_start_date
- address_end_date

Lookup Code
- code_type
- code_value
- code_description

Items in CAPITALS imply a standard o cross-reference.

13.7.8 Housing Services Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/enterprise_data_model_for_local_government/housing_services.htm

This model includes Haringey Council in London, UK.
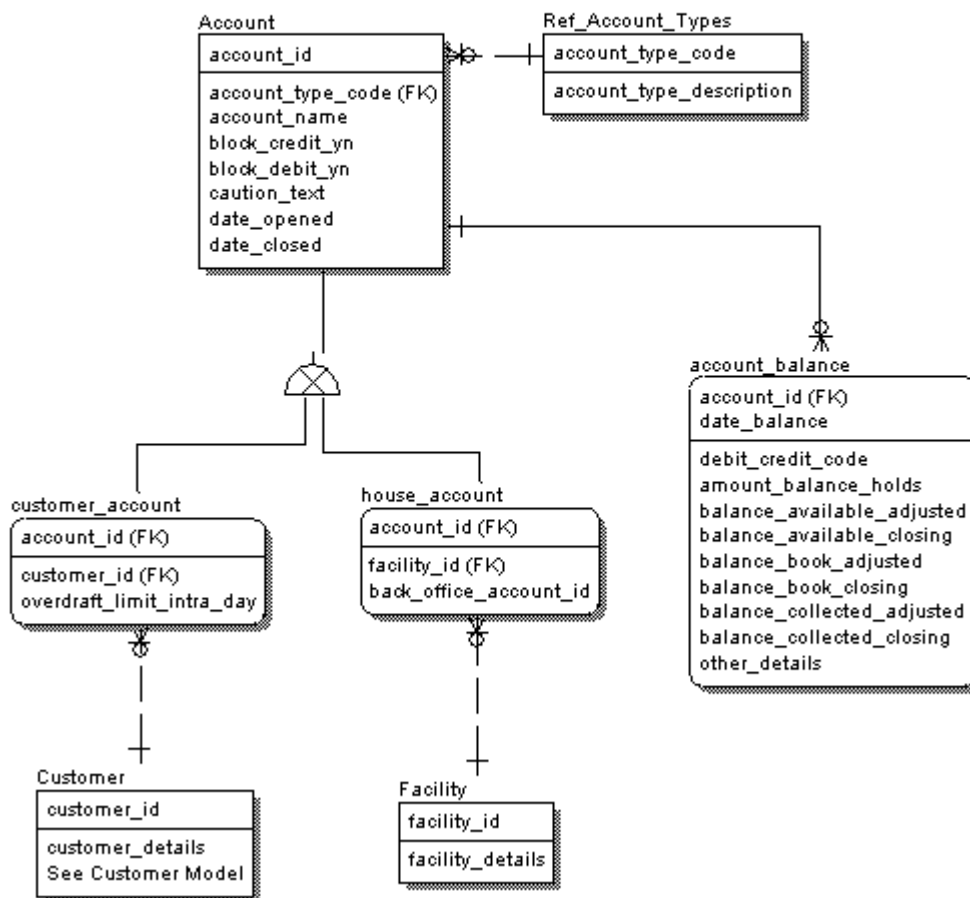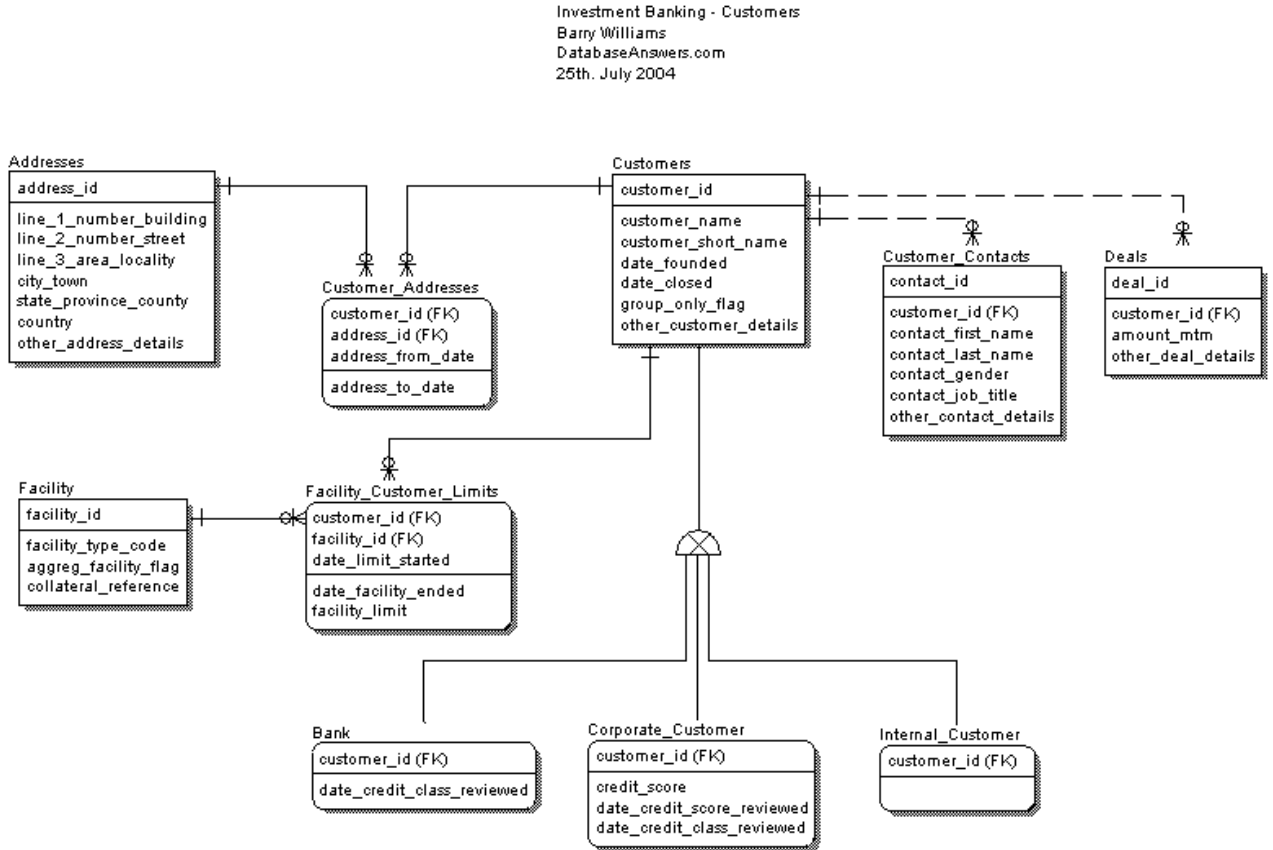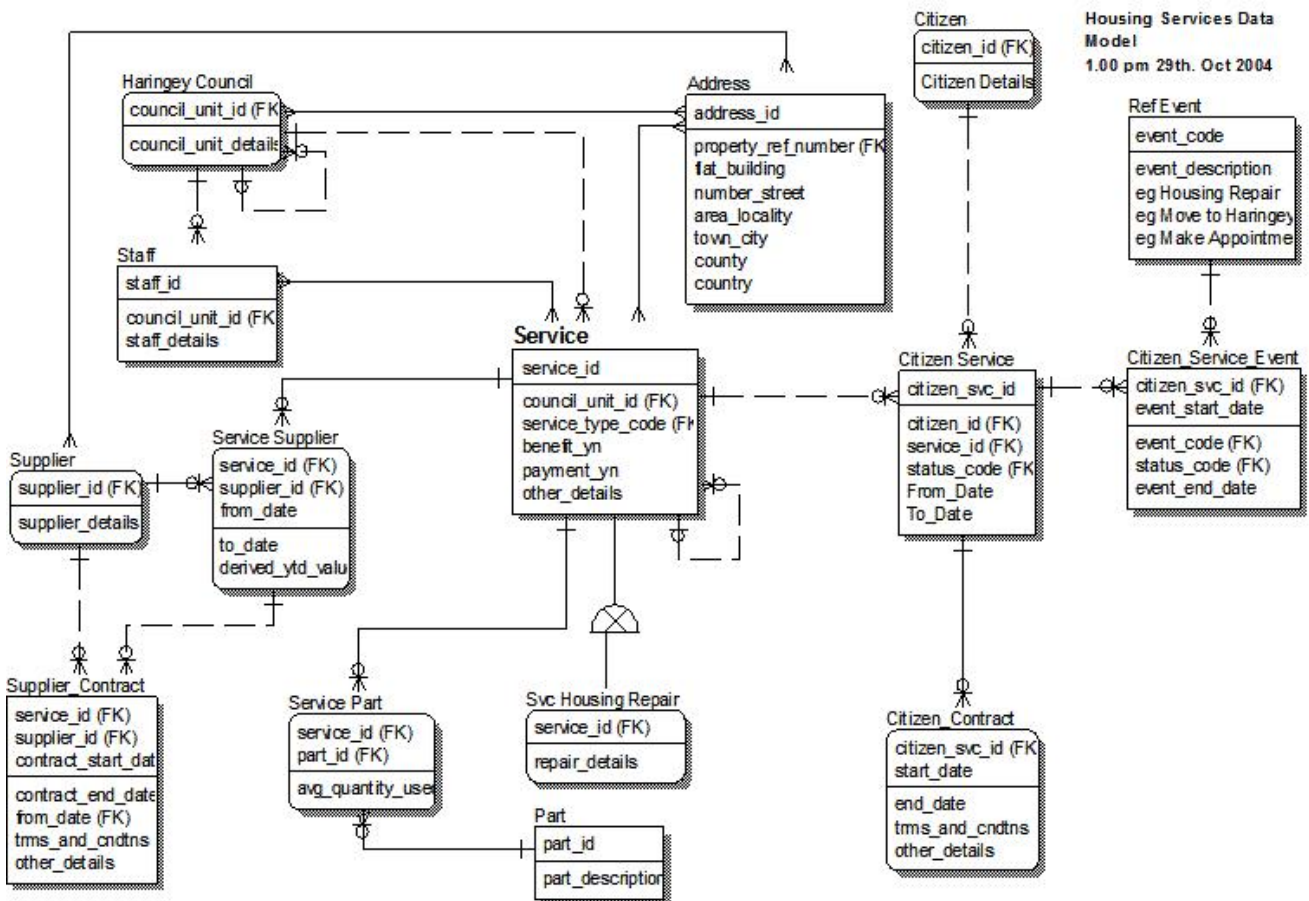
13.7.9 Data Warehouse

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/enterprise_data_model_for_local_government/data_warehouse.htm

Data Warehouse for Local Government
Barry Williams
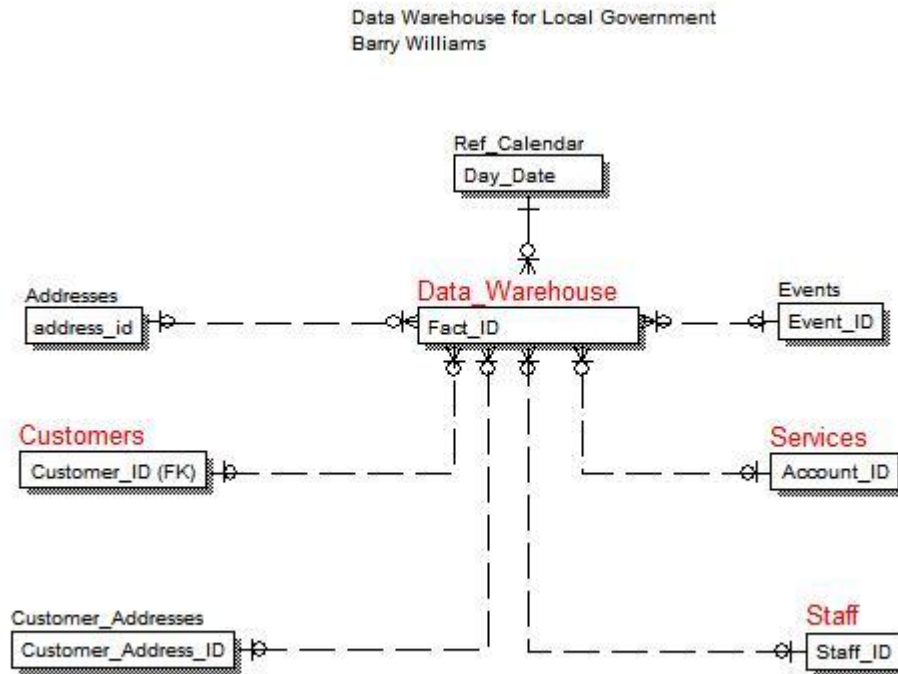
13.6 Retail

13.8.1 Statement of Objectives

The typical statement of objectives for a retail organization would be stated in these terms:

"To offer a range of affordably priced retail goods and services that are economically distributed and meet the requirements of a targeted customers. These will be made available in-store and online."

The subject areas reflect the statement of objectives. The purpose of this statement of objectives is to establish the scope of the enterprise data model:

The objectives are to offer a wide range of affordable products. Costs will be controlled to achieve maximum profit potential from a wide range of affordable products that are sourced on a 'just-in-time' basis tailored to our target market, with specific customer profiles.

The enterprise data model will establish the data available for a data warehouse to meet business intelligence requirements.

Therefore, the subject areas in the enterprise data model must include:

- Customers

- Distribution

- Inventory

- Marketing

- Products

13.8.2 Enterprise Data Model

This model appears on this page on the Database Answers Web site:

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

Enterprise Data Model for Retail
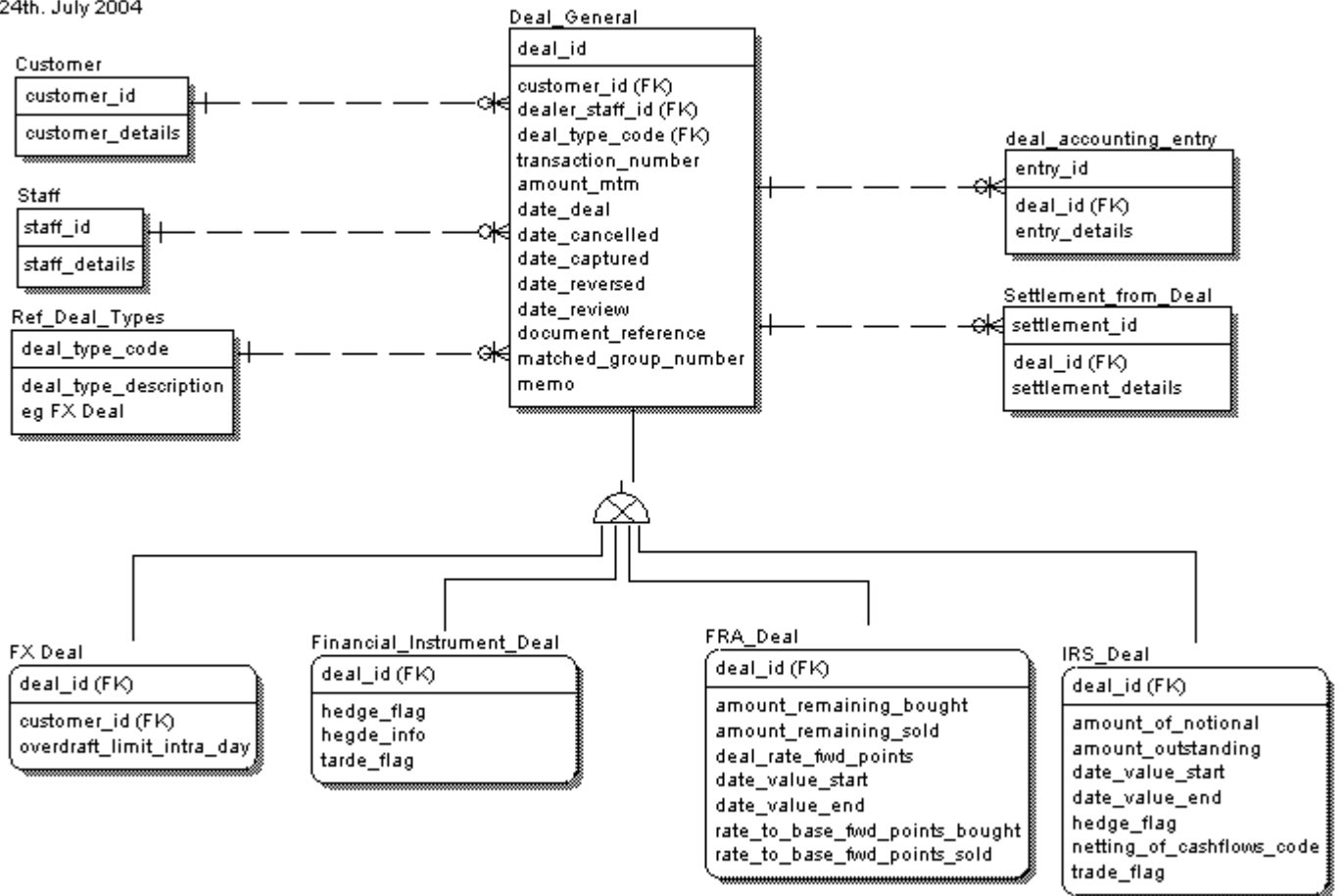Barry Williams
DatabaseAnswers.org
March 26th. 2011
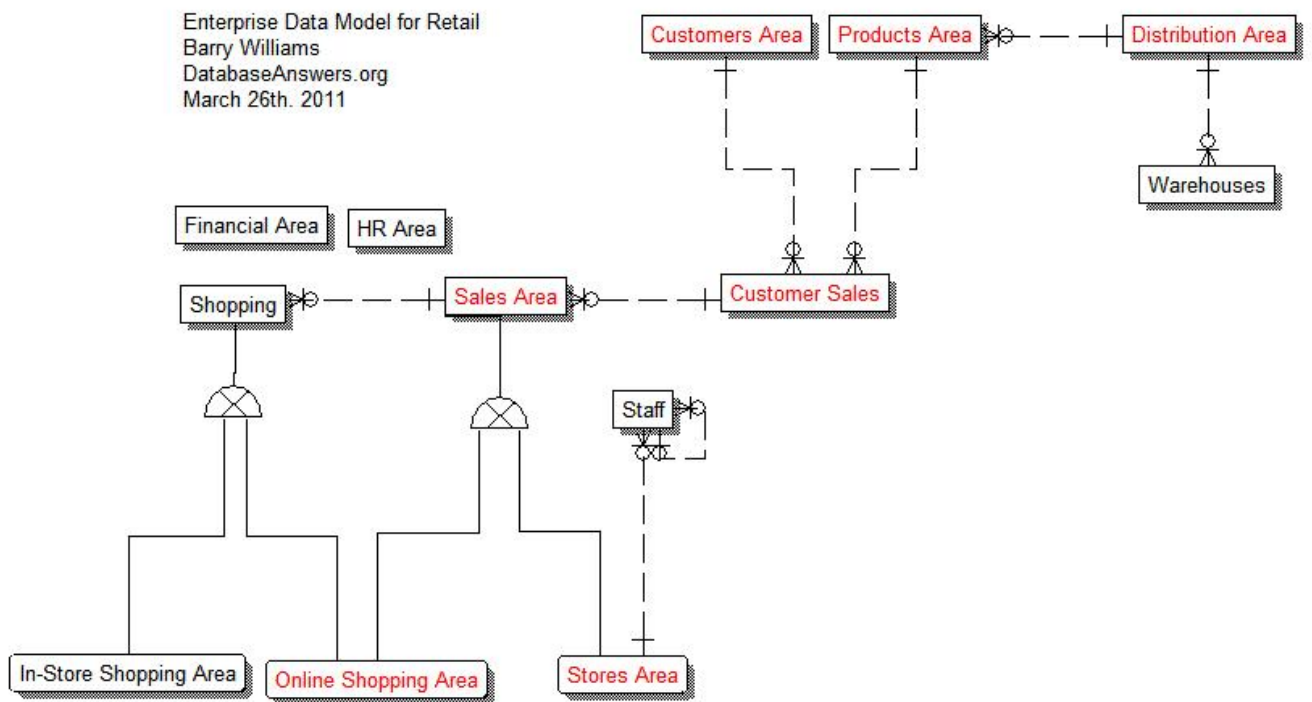
13.8.3 Customers Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/enterprise_data_model_for_retail/customers_area_model.htm
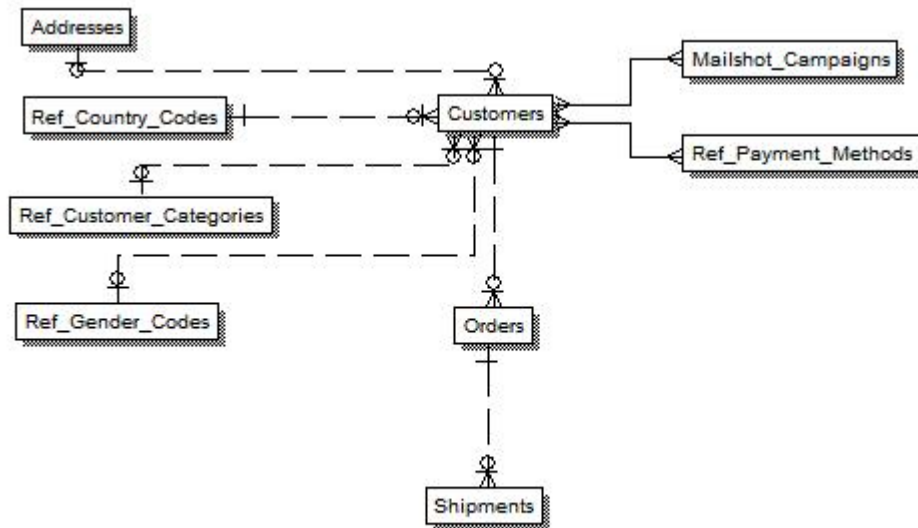
13.8.4 Customers and Orders Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/enterprise_data_model_for_retail/customers_and_orders_model.htm
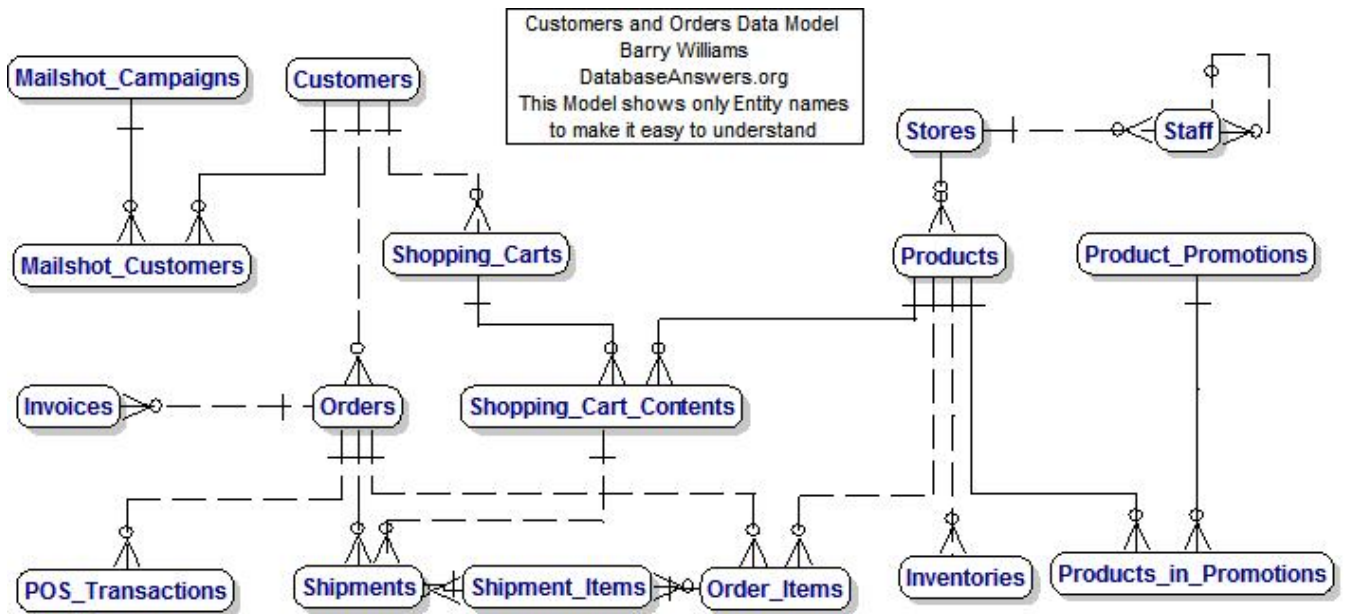
13.8.5 In-Store Shopping Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/enterprise_data_model_for_retail/in_store_shopping_area_model.htm
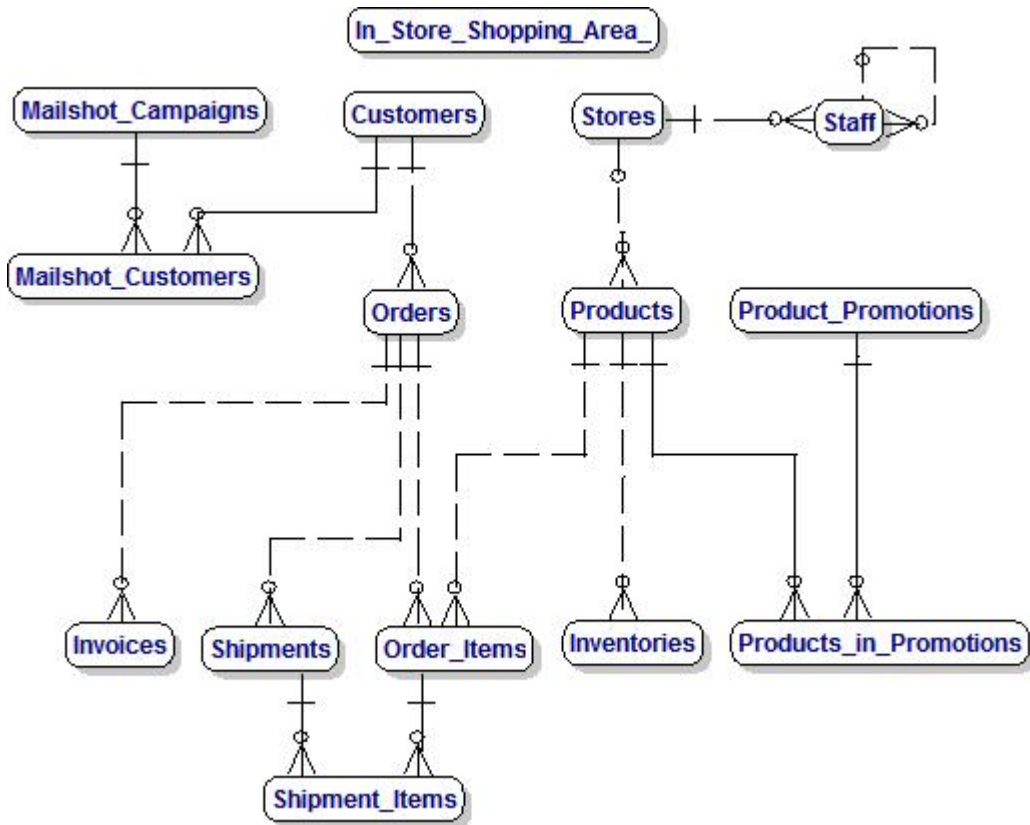
13.8.6 Online Shopping Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/enterprise_data_model_for_retail/online_shopping_area_model.htm
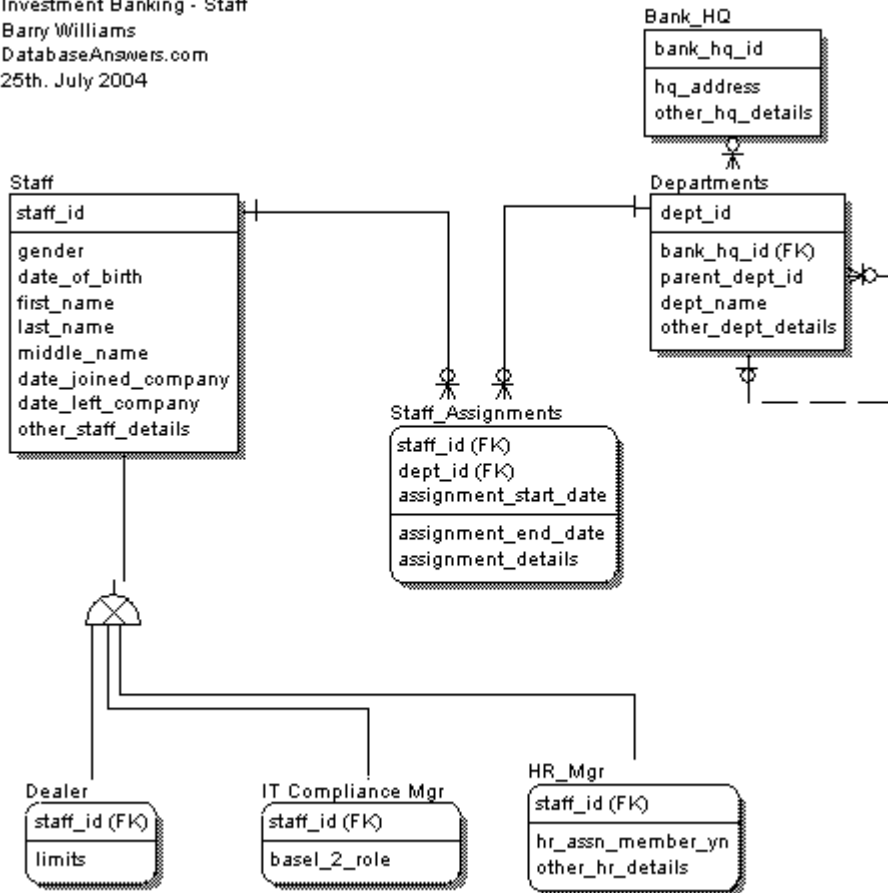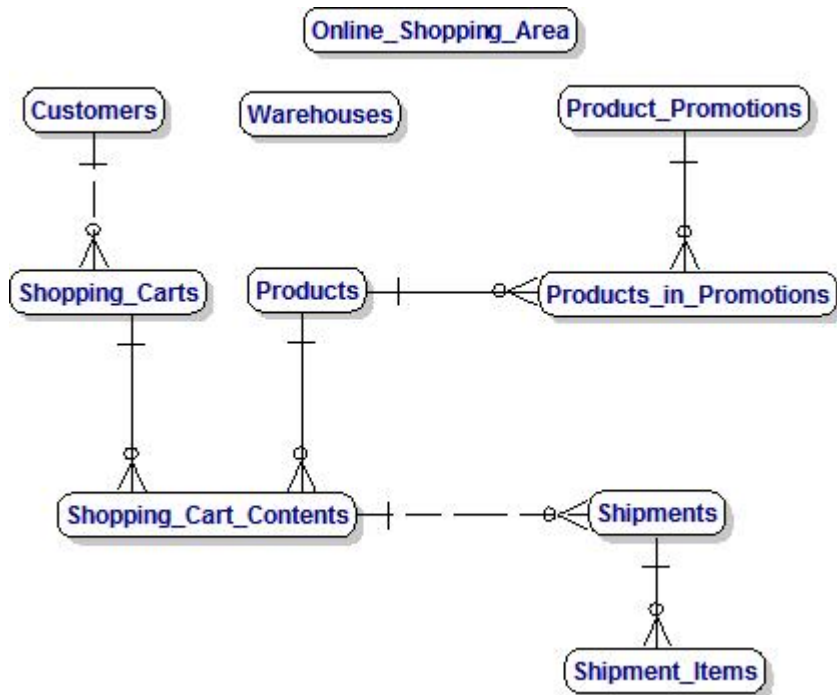
13.8.7 Order History Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/enterprise_data_model_for_retail/Order_History.htm
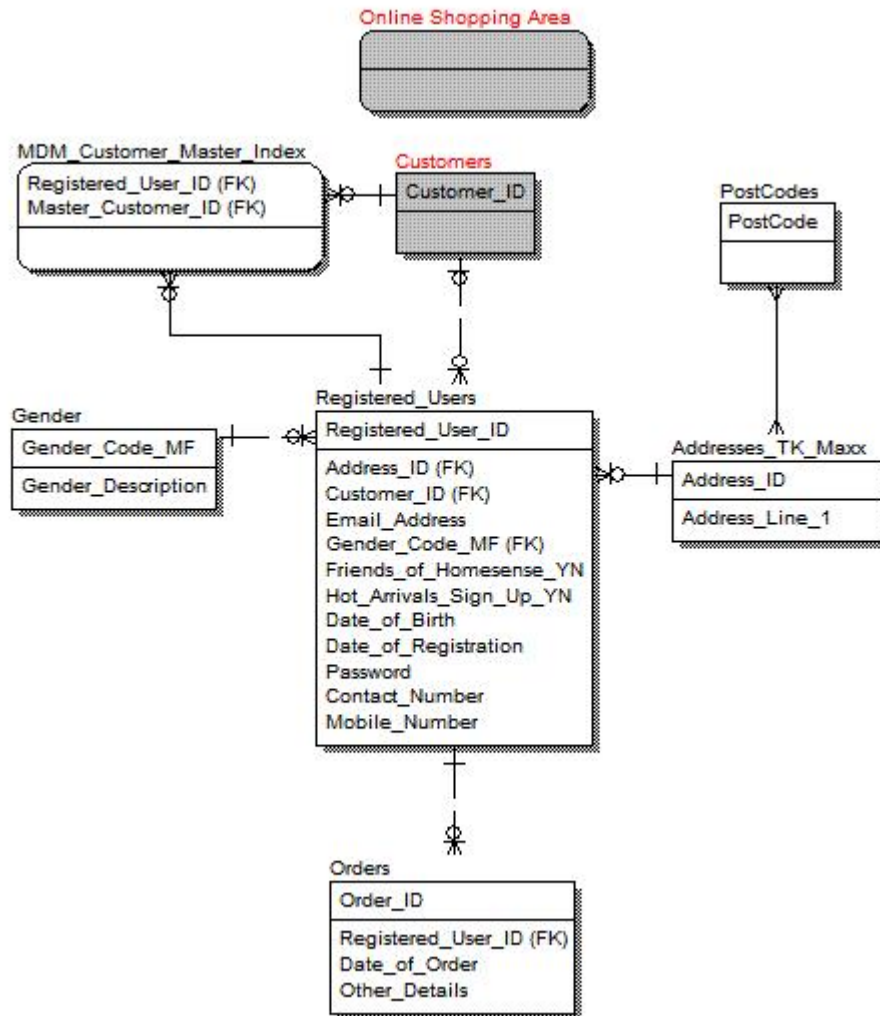
13.8.8 Products Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/enterprise_data_model_for_retail/Subject_Area_for_Products.htm
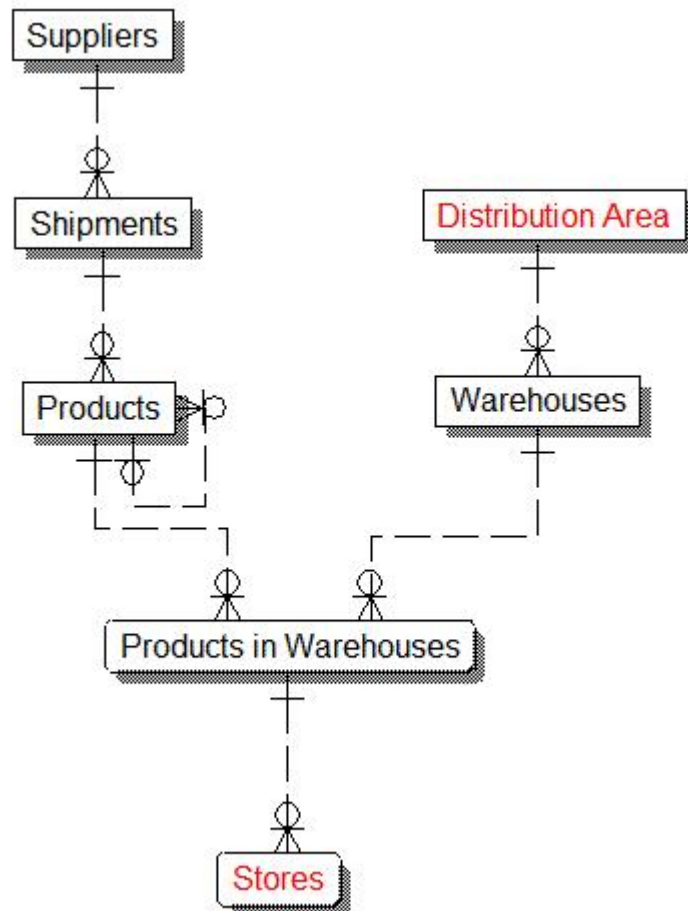
13.8.9 Products, Stores and Sales Subject Area

This model appears on this page on the Database Answers Web site:

- www.databaseanswers.org/data_models/enterprise_data_model_for_retail/Subject_Area_for_Products_Stores_and_Sales.htm

Products, Stores and Sales
Barry Williams
DatabaseAnswers.org
March 22nd. 2011
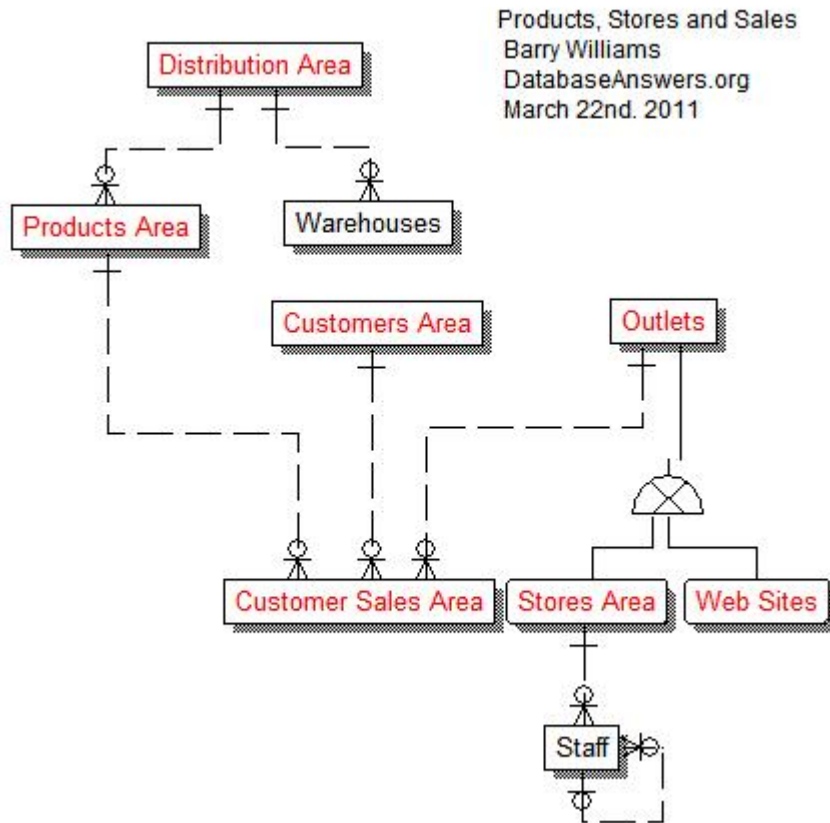
13.8.10 Registered Users (with MDM) Subject Area

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/enterprise_data_model_for_retail/Registered_Users.htm
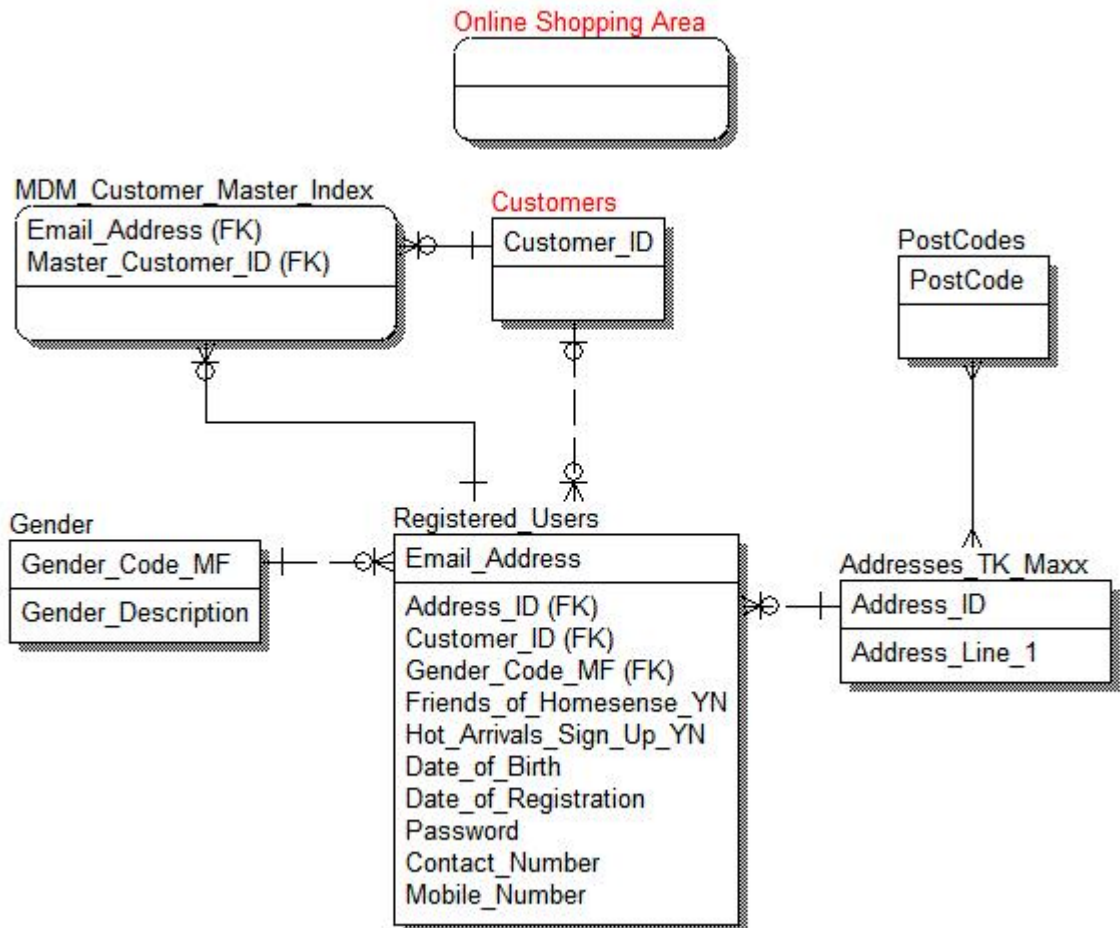
13.8.11 Data Warehouse

This model appears on this page on the Database Answers Web site:

- http://www.databaseanswers.org/data_models/enterprise_data_model_for_retail/retail_customers_data_mart.htm
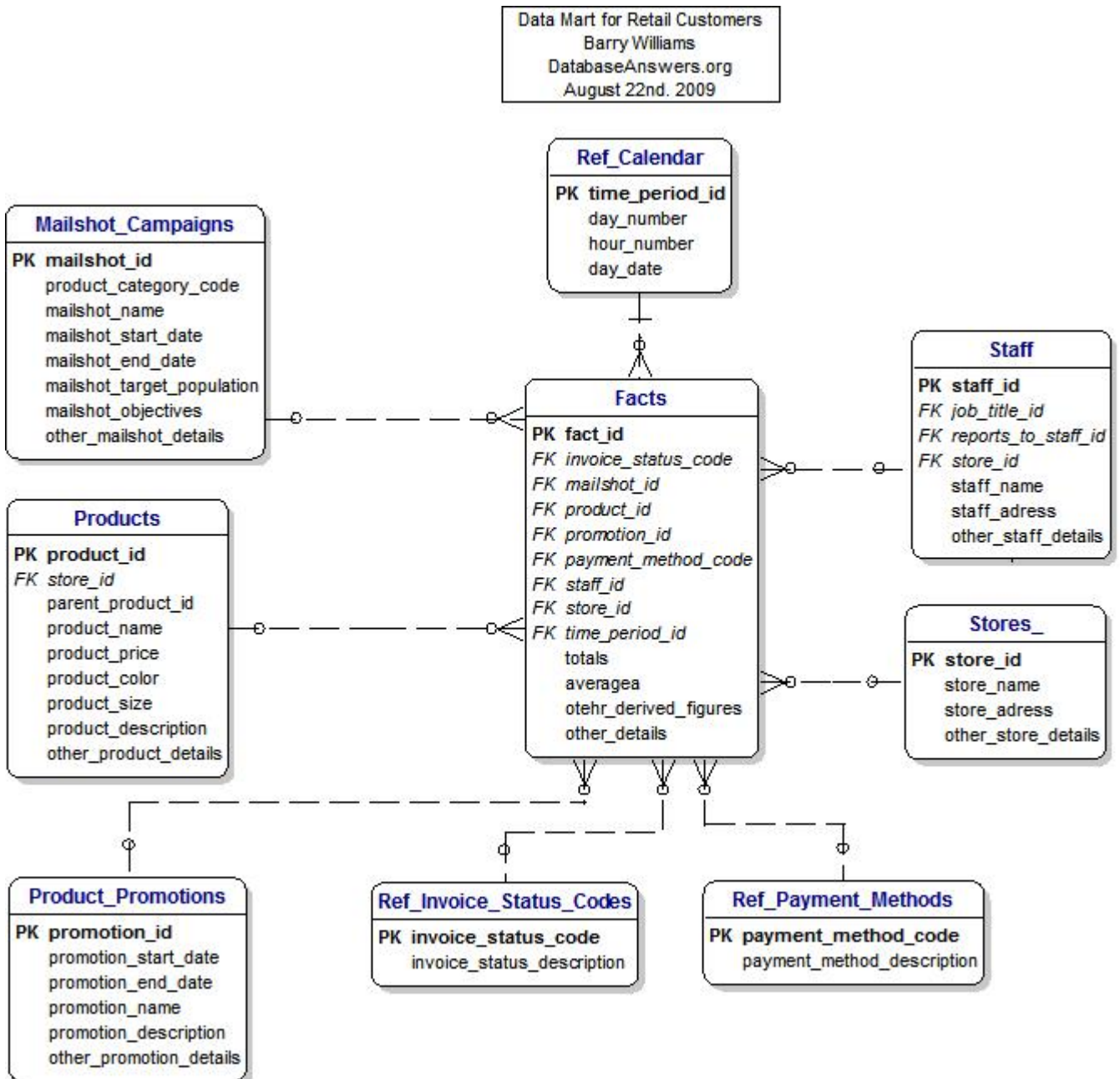
13.9 What have we learned?

In this chapter we have learned how to design data models for large commercial enterprises. These kind of enterprises are typically very complex and it can seem at first impossible to design a data model that is useful. In fact, it is common to see enterprise data models that include hundreds of entities. It is, of course, very difficult to use these models.

A good general approach is to start by designing a top-level data model with a limited number of entities. Then designing a number of subject area models will complement the top-level model to allow you to cover a wide range of functionality within the enterprise.

14. Master Data Management (MDM)

14.1 Introduction

This chapter will discuss **MDM** for CRM, law enforcement and local government.

14.1.1 What is this?

MDM is the common abbreviation for 'Master Data Management'.

Master data is the reference data, such as standard codes for countries, currencies, languages, products and so on.

14.1.2 Why is it important?

MDM is important because it is part of establishing a 'single view of the truth'.

14.1.3 What will I Learn?

In this chapter you will learn how to establish MDM, both from a data management point of view and from an organizational point of view.

14.1.4 Best Practice

Best practice for MDM establishes a 'single view of the truth'. It identifies the sources of data and determines how the sources are integrated and analyzes matching of reference data, products, customers and so on.

14.2 MDM for CRM

This section covers MDM for CRM.

During this short tutorial, we will cover the following topics:
1.   An approach to MDM
2.   Creating a data dictionary
3.   Mapping from sources to the target MDM data model.
4.   Reference data

This tutorial presents the steps involved as best practice. Topics covered in this tutorial include data sources and targets, mapping between them, data quality, data governance and reference data.

These are the steps in the tutorial:
1.   Agree the scope of the project
2.   Identify the data sources
3.   Determine the target
4.   Specify the mapping between sources and target.

5.   Review guidelines for managing reference data
6.   Address the issue of data quality
7.   Ensure compliance with data governance
8.   Check that the MDM data model can provide data for performance reports.

14.2.1 Top-Level CRM Data Model

This is the MDM CRM data model that is the target for data migration.
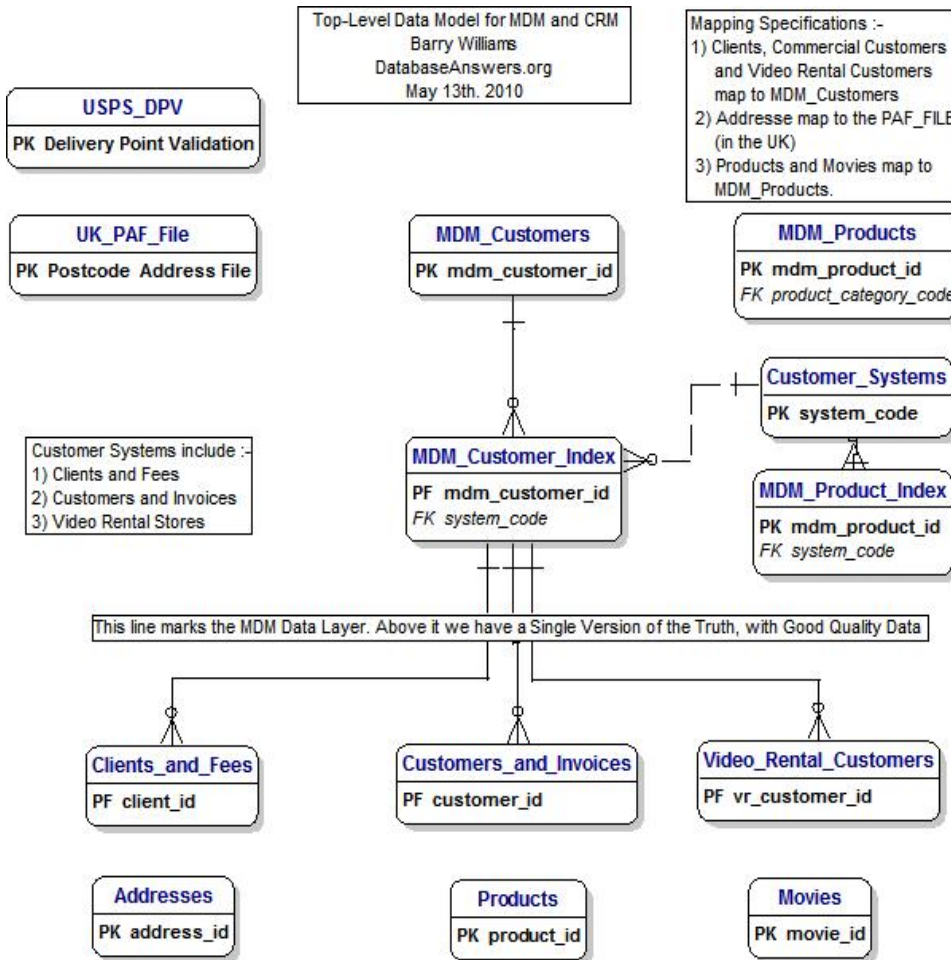
14.2.2 Reference Data Model

Top-Level Data Model for MDM and CRM
Barry Williams
DatabaseAnswers.org
May 9th. 2010

Ref_Data_Types

PK data_type_code
   data_type_name
   eg Movies

Ref_Data

PF data_type_code
PK ref_data_code
   ref_data_name
   ref_data_description
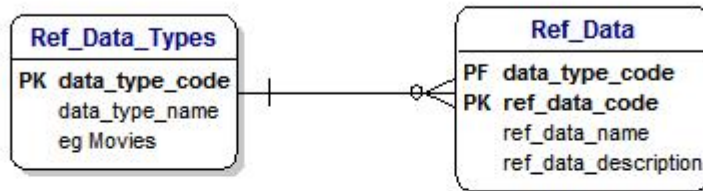
This line marks the MDM Data Layer. Above it we have a Single Version of the Truth, with Good Quality Data

Ref Data for Clients and Fees   Ref Data for Clustomers and Invoices   Ref Data for Video Rentals

Ref_Calendar

PK day_date
   day_number

Ref_Invoice_Status

PK invoice_status_code
   invoice_status_description

Ref_Job_Types

PK job_type_code
   job_type_description

Ref_Skills

PK skill_code
   skill_description

Ref_Skill_Levels

PK skill_level_code
   skill_level_description

Commercial_Product_Categories

PK product_category_code
FK *parent_product_category_code*
   product_category_name
   product_category_description
   eg Cl;othing, Food and Hardware

Transaction_Types

PK transaction_type_code
   transaction_type_description
   eg Adjustment
   eg Payment, Refund

Movie_Format_Types

PK format_type_code
   format_type_name
   format_type description
   eg DVD

Movie_Genre_Codes

PK genre_type_code
   genre_type_name
   genre_type description
   eg Comedy, Western

Payment_Methods

PK payment_method_code
   payment_method_name
   eg Amex, Diners Club
   eg MasterCard, Visa

Rental_Status_Codes

PK rental_status_code
   rental_status_description
   eg Booked, Out, Returned

Video_Transaction_Types

PK transaction_type_code
   transaction_type_description
   eg Payment, Refund

14.2.3 Clients and Fees

Client_Addresses
- PF client_id*
- PF address_id*
- PK date_address_from*
- date_address_to

Addresses
- PK address_id*
- line_1_number_building
- line_2_number_street
- line_3_area_locality
- town_city
- state_province
- country_code

Staff_Addresses
- PF staff_id*
- PF address_id*
- PK date_address_from*
- date_address_to

Ref_Job_Types
- PK job_type_code*
- job_type_name

Clients
- PK client_id*
- client_name
- client_from_date
- kpi_avg_billable_rate
- kpi_billings_to_date
- kpi_client_project_count
- other_client_details

Projects
- PK project_id*
- FK client_id*
- project_name
- project_start_date
- project_end_date
- project_description
- other_project_details

Staff
- PK staff_id*
- FK job_type_code*
- staff_name
- other_staff_details

Ref_Skills
- PK skill_code*
- skill_name
- eg Analyst

Fees
- PF client_id*
- PF period_id*
- total_fees

Ref_Calendar
- PK day_date*
- business_day_yn
- day_number
- period_id

Staff_on_Projects
- PK staff_on_project_period_id*
- FK project_id*
- FK staff_id*
- FK role_code*
- from_datetime
- to_datetime
- hourly_rate

Staff_Skills
- PF staff_id*
- PF skill_code*
- FK skill_level_code

Ref_Skill_Levels
- PK skill_level_code*
- skill_level_name
- eg Advanced, 5 yrs exp..

Invoices
- PK invoice_id*
- invoice_number
- FK period_id*
- FK client_id*
- FK invoice_status_code*
- date_issued
- date_paid
- total_fees
- discount
- amount_payable
- other_details

Ref_Invoice_Status
- PK invoice_status_code*
- invoice_status_description
- eg Cancelled,Disputed,Paid.

Ref_Roles
- PK role_code*
- role_name
- eg Project Manager

Clients and Fees
Barry Williams
14th. April 2004
DatabaseAnswers.org

14.2.4 Customers and Invoices

This model shows that customers receive invoices for orders that they place.

Customers
- PK **customer_id***
- customer_first_name
- customer_middle_initial
- customer_last_name
- gender
- email_address
- login_name
- login_password
- phone_number
- address_line_1
- address_line_2
- address_line_3
- address_line_4
- town_city
- state_county_province
- country
- other_details

Customers and Invoices Data Model
Barry Williams
30th. Jan 2005
DatabaseAnswers.org

Accounts
- PK **account_id***
- FK *customer_id**
- date_account_opened
- account_name
- other_account_details

Orders
- PK **order_id***
- FK *customer_id**
- date_order_placed*
- order_details

Invoices
- PK **invoice_number***
- FK *order_id**
- invoice_date
- invoice_details

Products
- PK **product_id***
- FK *production_type_code**
- product_name
- product_price
- product_color
- product_size
- product_description
- other_product_details

Order_Items
- PK **order_item_id***
- FK *order_id**
- FK *product_id**
- product_quantity
- other_order_item_details

Invoice_Line_Items
- PF **order_item_id***
- FK *invoice_number**
- FK *product_id**
- product_title
- product_quantity
- product_price
- derived_product_cost
- derived_vat_payable
- derived_total_cost
- other_line_item_details

Financial_Transactions
- PK **transaction_id***
- FK *account_id**
- FK *invoice_number**
- FK *transaction_type_code**
- transaction_date
- transaction_amount
- transaction_comment
- other_transaction_details

Product_Types
- PK **production_type_code***
- FK *parent_production_type_code*
- product_type_description
- vat_rating
- eg Clothing, Food, Hardware.

Transaction_Types
- PK **transaction_type_code***
- transaction_type_description
- eg Payment, Refund.

14.2.5 Video Rental Stores

Data Model for Video Rental Stores
Barry Williams
DatabaseAnswers.org
29th. Sept 2006

Video_Stores

PK store_id

    store_name
    store_address
    store_phone
    store_email
    other_store_details

Rental_Status_Codes

PK rental_status_code

    rental_status_description

Condition_Codes

PK condition_code

    condition_description

Movies

PK movie_id

FK condition_code
FK format_type_code
FK genre_type_code
FK store_id
    release_year
    movie_title
    movie_description
    number_in_stock
    rental_or_sale_or_both
    rental_daily_rate
    sale_price

Customers

PK customer_id

    member_yn
    membership_number
    date_became_member
    customer_first_name
    customer_last_name
    customer_address
    customer_phone
    customer_email
    other_customer_details

Format_Types

PK format_type_code

    format_type_description

Customer_Rentals

PK item_rental_id

FK customer_id
FK movie_id
FK rental_status_code
    rental_date_out
    rental_date_returned
    rental_amount_due
    other_rental_details

Genre_Codes

PK genre_code

    genre_description

Accounts

PK account_id

FK customer_id
FK payment_method_code
    account_name
    account_details

Actors

PK actor_id

    actor_gender
    actor_first_name
    actor_last_name
    other_actor_details

Movie_Cast

PF movie_id
PF actor_id

Financial_Transactions

PK transaction_id

FK account_id
FK item_rental_id
FK previous_transaction_id
FK transaction_type_code
    transaction_date
    transaction_amount
    transaction_comment

Payment_Methods

PK payment_method_code

    payment_method_description
    eg Amex, Cash, CC

Transaction_Types

PK transaction_type_code

    transaction_type_description
    eg Payment, Refund

14.3 MDM for Law Enforcement

This chapter covers **Master Data Management** for law enforcement. It presents the steps involved as best practice.

Topics covered in this tutorial include data sources and targets, mapping between them, data quality, data governance and reference data.

I hope you find this tutorial interesting and helpful.

Please email me and let me know at barryw@databaseanswers.org.

These are the steps in the tutorial:

1.   Agree the scope of the project
2.   Identify the data sources
3.   Determine the target
4.   Specify the mapping between sources and target.
5.   Review guidelines for managing reference data
6.   Address the issue of data quality
7.   Ensure compliance with data governance
8.   Check that the MDM data model can provide data for Police Information Reports and the National Intelligence Model.

The starting point is a list of relevant Data Models for Law Enforcement.

14.3.1 Top-Level Police Data Model

This is the MDM police data model, which is the target. It shows how data from a number of sources is integrated into one generic master data management model.

NOTES
Violaters are incorporated into MDM_Persons_of_Interest.
Violater Addresses into MDM_PAF_Addresses.
Ref_Violation_Types into MDM_Offence_Categories.

Police Master Data Management Data Model
Barry Williams
May 9th. 2010

14.3.2 Reference Data

Each Data Source will have its own sets of reference data.
These have to be mapped to a common set of data, which in turn is subject to corporate data governance.

14.3.3 Case Management

This model shows that cases go to trial where people are in attendance, playing specific roles.

14.3.4 Police Departments

This model shows that incidents are central to a police department and people are involved in incidents, playing specific roles.

Data Model for Police Departments
Barry Williams
10th. February 2009
DatabaseAnswers.org

14.3.5 Prisons and Prisoners

This model shows that prisoners commit offences and have associates.

Data Model for Prisons and Prisoners
Barry Williams
DatabaseAnswers.org
17th. July 2004

Ref_Staff_Roles
- **PK** **staff_role_code**
- staff_role_description
- - eg Governor, Warder.

Prison_Staff
- **PK** **prison_staff_id**
- **FK** *prison_id*
- **FK** *staff_id*
- **FK** *staff_role_code*
- role_start_date
- role_end_date
- other_role_details

Prisons
- **PK** **prison_id**
- prison_name
- other_prison_details
- prison_governer_staff_id

Ref_Offence_Categories
- **PK** **offence_category_code**
- offence_category_description
- - eg Felony, Misdemeanour.

Ref_Probation_Status
- **PK** **probation_status_code**
- probation_status_description
- - eg Approved, Rejected.

Associates
- **PK** **associate_id**
- associate_first_name
- associate_middle_name
- associate_last_name
- other_associate_details

Prisoners
- **PK** **prisoner_id**
- **FK** *gender_code*
- **FK** *prison_id*
- **FK** *probation_status_code*
- prisoner_first_name
- prisoner_middle_name
- prisoner_last_name
- prison_number
- prisoner_address
- probation_hearing_due_date
- other_prisoner_details

Offences
- **PK** **offence_id**
- **FK** *offence_category_code*
- offence_short_name
- offence_full_name
- offence_description
- offence_min_sentence
- offence_max_sentence
- other_offence_details

Prisoner_Associates
- **PF** **prisoner_id**
- **PF** **associate_id**
- associated_from_date
- last_known_contact_date
- last_known_contact_location

Prisoner_Offences
- **PK** **prisoner_offence_id**
- **FK** *prisoner_id*
- **FK** *offence_id*
- datetime_committed
- location_committed

Prisoner_Associate_Visits
- **PF** **prisoner_id**
- **PF** **associate_id**
- **PK** **visit_start_datetime**
- visit_end_datetime
- comments

14.3.6 Tracking Evidence

This model shows that officers track evidence for cases and maintain a chain of custody.

Diary_of_Events

- PK **event_date***
- FK *individual_id*
- active_to_date
- event_description*

Ref_People_Roles

- PK **role_code***
- - eg SUS1, VIC1.
- role_description
- - eg Suspect 1, Victim 1.

Ref_Title

- PK **title_code***
- - eg Mr.,Mrs.,Ms..
- title_description

People

- PK **person_id***
- FK *address_id*
- FK *role_code*
- FK *title_code*
- first_name
- middle_name
- last_name
- date_of_birth
- date_in
- date_out
- phone_number
- cell_phone
- email_address
- comments
- other_person_details

Case_Files

- PK **file_number***
- FK *investigating_officer_id*
- control_number
- date_in
- date_out
- charges

Tracking Evidence
Barry Williams
24th. February 2004
DatabaseAnswers.org

Addresses

- PK **address_id***
- line_1_number_building
- line_2_number_street
- line_3_area_locality
- city
- zip_postcode
- state_province_county
- country
- other_address_details

Ref_Ranks

- PK **rank_code***
- rank_short_name
- - eg Sgt., Officer, Lt.
- rank_description

Ref_Shifts

- PK **shift_code***
- shift_description
- eg B Watch, Traffic.

Officers

- PF **officer_id***
- FK *dept_code**
- FK *rank_code**
- FK *shift_code**
- badge_number
- date_became_officer
- other_officer_details

Departments

- PK **dept_code***
- FK *parent_dept_code*
- dept_description

Evidence_Items

- PK **item_number***
- FK *category_code*
- FK *file_number**
- FK *officer_id**
- FK *location_code**
- sealed_yn
- date_item_in
- date_item_out
- item_origin
- item_disposal
- item_description
- item_comments
- other_item_details

Ref_Item_Categories

- PK **category_code***
- category_description
- - eg Knife, Firearm

Ref_Locations

- PK **location_code***
- location_description
- - eg Aisle 2, Shelf B

Chain_of_Custody

- PF **item_number***
- PF **moved_from_location_code***
- FK *moved_to_location_code**
- date_moved
- time_moved

14.3.7 Traffic Cops and Tickets

This model shows that traffic cops, who are officers, issue tickets for violations. These tickets are defined as specific types of documents.

Ref_Payment_Methods
PK payment_method_code
   payment_method_description

Violaters
PK violater_id
FK violater_address_id
FK payment_method_code
   violater_name
   violater_phone
   violater_email
   date_of_first_violation
   other_violater_details

Addresses
PK address_id
   line_1_number_building
   line_2_number_street
   line_3_area_locality_suburb
   line_4_other
   town_city
   zip_postcode
   state_province_county
   country
   other_address_details

Ref_Vehicle_Manufacturers
PK manufacturer_code
   manufacturer_name

Vehicles
PK vehicle_licence_number
FK manufacturer_code
FK vehicle_type_code
   year_of_manufacture
   vehicle_details
   model

Violater_Vehicles
PF violater_id
PF vehicle_licence_number

Officers
PK officer_badge_number
FK officer_address_id
   gender
   date_of_birth
   officer_first_name
   officer_middle_name
   officer_last_name
   other_officer_details

Violations
PK violation_id
FK violater_id
FK vehicle_licence_number
FK document_type_code
FK officer_badge_number
   datetime_violation
   datetime_order_delivered
   other_offence_details

Ref_Vehicle_Types
PK vehicle_type_code
   vehicle_type_description
   eg SUV

Ref_Document_Types
PK document_type_code
   documentation_type_description

Tickets
PK ticket_number
FK ticket_status_code
FK violation_id
   date_ticket_cancelled
   date_ticket_paid
   other_ticket_details

Ref_Ticket_Status
PK ticket_status_code
   ticket_status_description
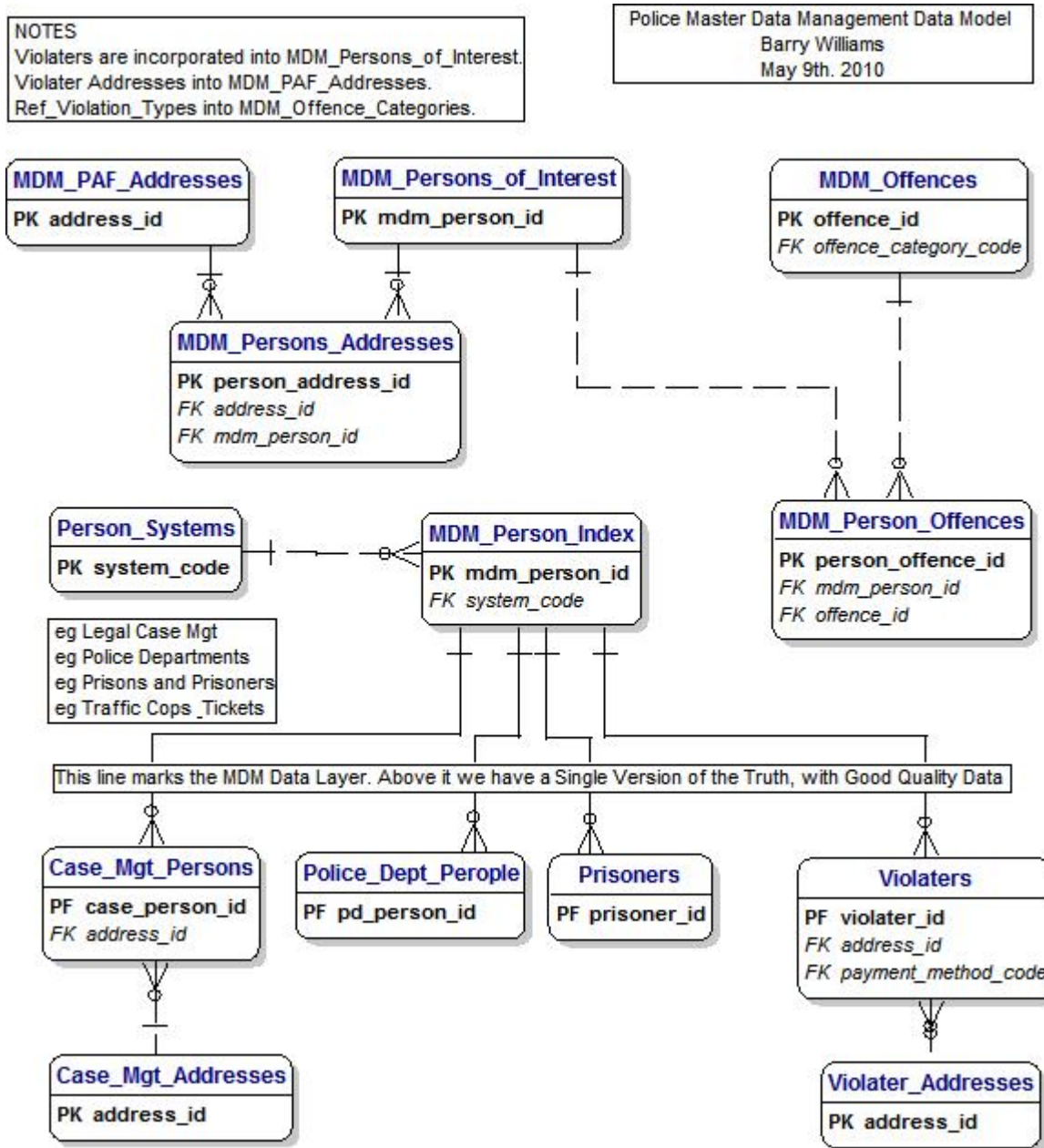   eg Cancelled, Issued, Paid

14.4 MDM for Local Government

These are the steps in the best practice:

1. Agree the scope of the project
2. Identify the data sources
3. Determine the target
4. Review reference data

14.4.1 Top-Level Data Model

This is the MDM Local Government data model, which is the target. This shows how data from a number of sources is integrated into one generic MDM model.
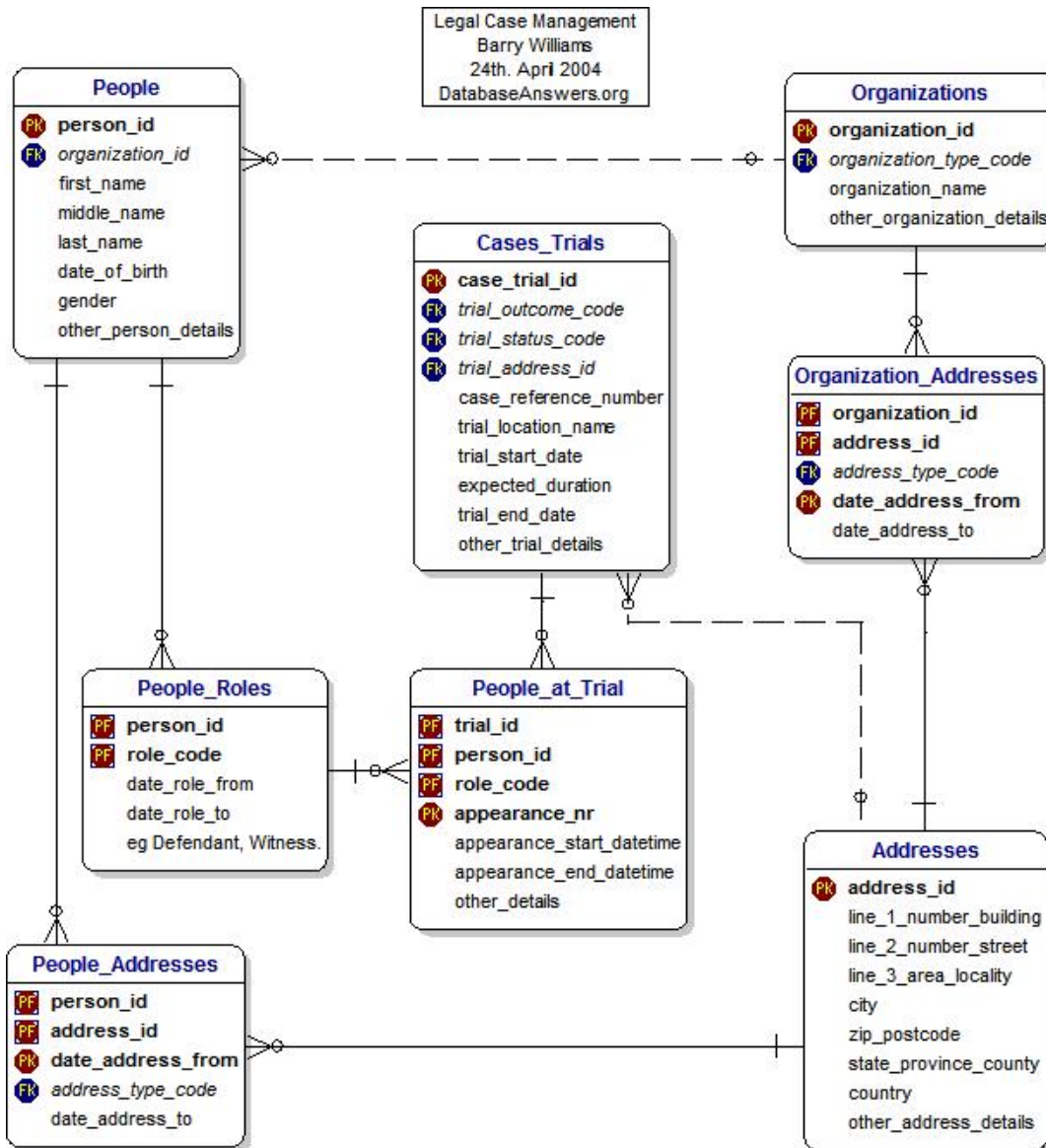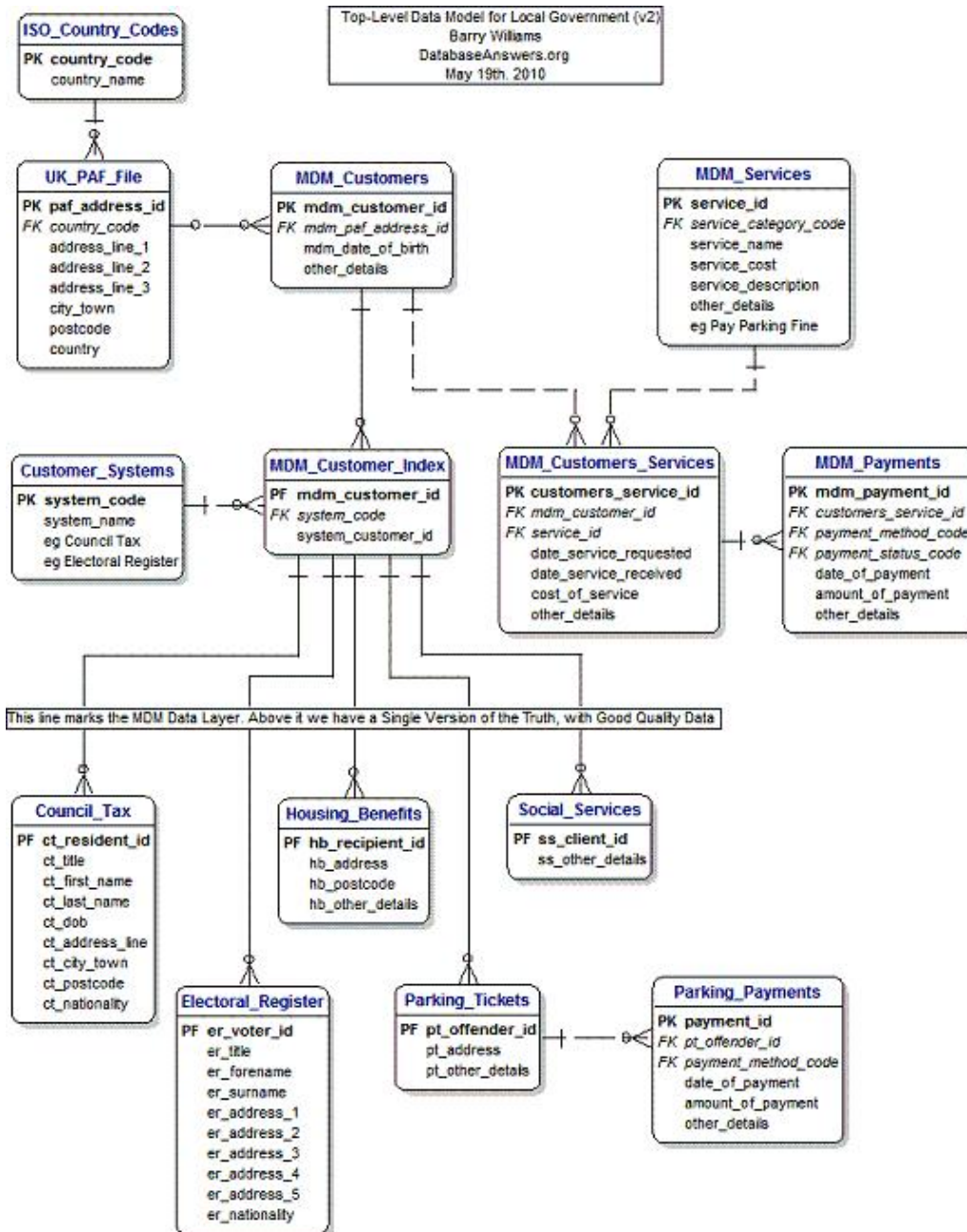
14.4.2 Reference Data

Each data source will have its own sets of reference data. These have to be mapped to a common set of data, which are subject to corporate Data Governance.
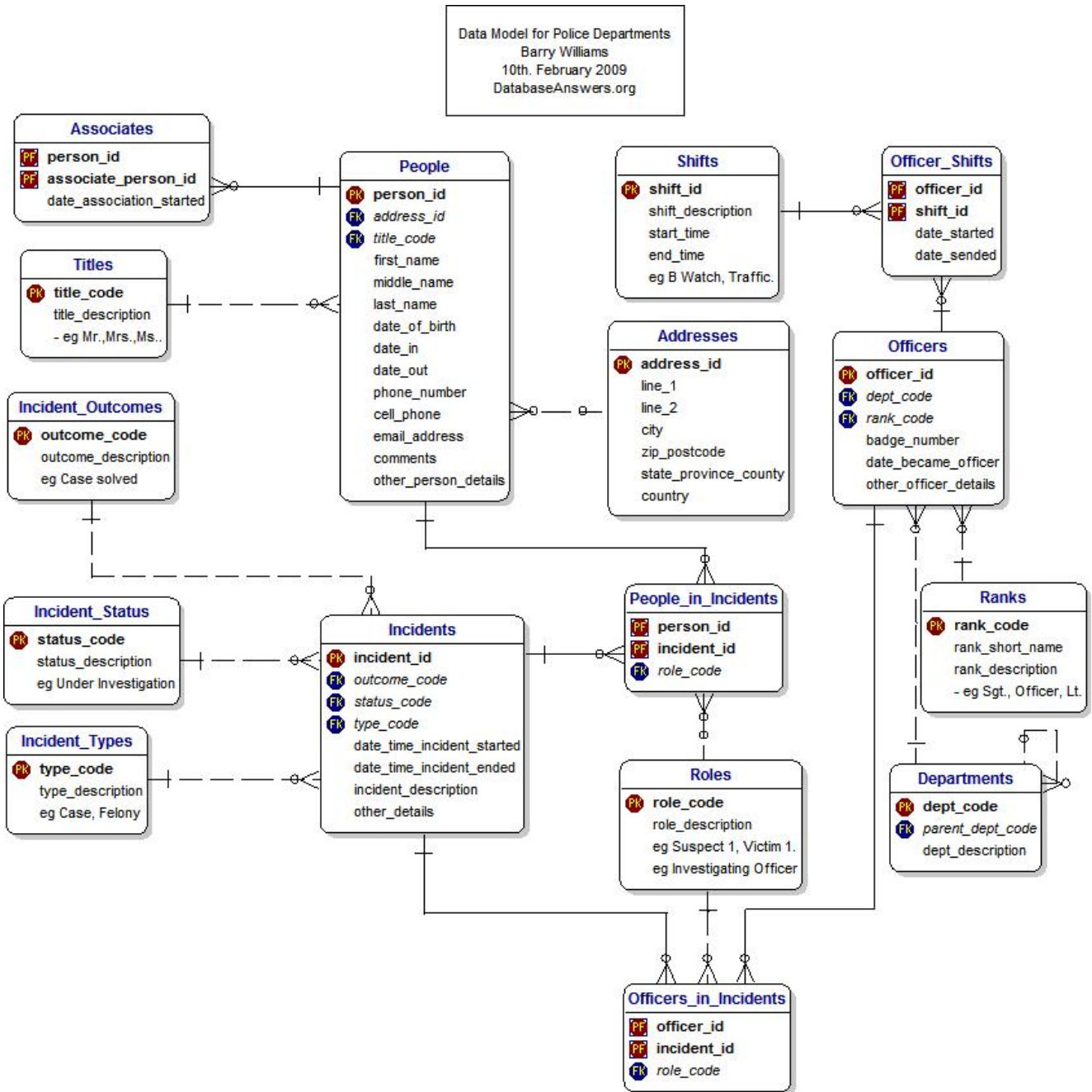
Ref_Data_Types

PK data_type_code
  data_type_name
  eg Country Codes
  eg Language Codes

Ref_Data

PF data_type_code
PK ref_data_code
  ref_data_name
  eg Cash, Credit Card

Ref_ISO_Language_Codes

PK iso_language_code
  iso_language_name
  eg EN=English

Ref_ISO_Country_Codes

PK iso_country_code
  iso_country_name
  eg UK=United Kingdom

Ref_Service_Categories

PK service_category_code
FK parent_service_category_code
  service_category_name
  eg Community Centres and Halls
  eg Council Tax, Electoral Register
  eg Housing Benefits, Parking
  eg Scial Services

Ref_UK_PAF_File

PK paf_address_id
FK country_code
  address_line_1
  address_line_2
  address_line_3
  city_town
  postcode
  country

Ref_Payment_Methods

PK payment_method_code
  payment_method_name
  eg Amex, Diners Club
  eg MasterCard, Visa

14.4.3 MDM and Customers and Services

This is a generic model that shows a service hierarchy where a many-to-many relationship between customers and services has been resolved into two one-to-many relationships.

MDM Customers and Services
Barry Williams
DatabaseAnswers.org
May 16th. 2010

Ref_Service_Categories

PK **service_category_code**

FK *parent_service_category_code*
   service_category_name
   eg Community Centres and Halls
   eg Council Tax, Electoral Register
   eg Housing Benefits, Parking
   eg Scial Services

MDM_Customers

PK **mdm_customer_id**

FK *iso_country_of_origin_code*
FK *iso_preferred_language_code*
FK *mdm_paf_address_id*
   mdm_personal_name
   mdm_middle_name
   mdm_family_name
   mdm_alias
   mdm_date_of_birth
   mdm_gender_mfu
   mdm_other_details

MDM_Services

PK **service_id**

FK *service_category_code*
   service_name
   service_cost
   service_description
   other_details
   eg Pay Parking Fine

MDM_Customers_Services

PK **customers_service_id**

FK *mdm_customer_id*
FK *service_id*
   date_service_requested
   date_service_received
   cost_of_service
   other_details

14.4.4 Council Tax

This shows the fields in the tables that are candidates for mapping to the MDM tables.

```
Data Model for Council Tax
       Barry Wlliams
   DatabaseAnswers.org
      May 19th. 2010
```

```
        Council_Tax
PF  ct_resident_id
    ct_title
    ct_first_name
    ct_last_name
    ct_dob
    ct_address_line
    ct_city_town
    ct_postcode
    ct_nationality
```

14.4.5 Housing Benefits

This shows the fields in the tables that are candidates for mapping to the MDM tables.

```
Data Model for Housing Benefits
        Barry Williams
    DatabaseAnswers.org
       May 16th. 2010
```

```
      Housing_Benefits
PF  hb_recipient_id
    hb_address
    hb_postcode
    hb_name
    hb_alias
    hb_other_details
```

14.4.6 Parking Tickets

This shows the fields in the tables that are candidates for mapping to the MDM tables.

Parking tickets are not master data and are not included in this mapping activity. However, the payment methods are master data and they are included in the reference data category.
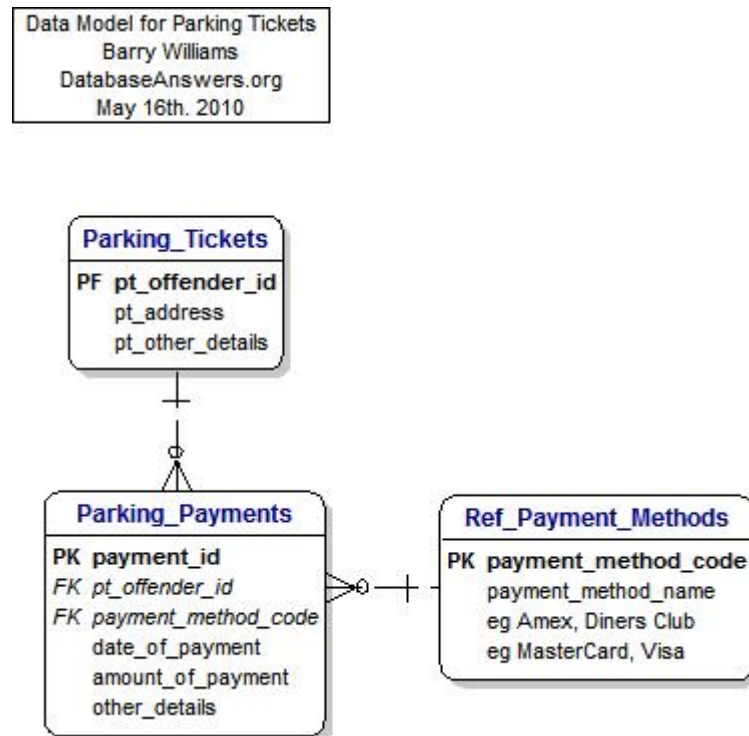
14.4.7 Social Services

This shows the fields in the tables that are candidates for mapping to the MDM tables. The structure of the Social Services Table must be evaluated as candidates for migration to the MDM Services Table.
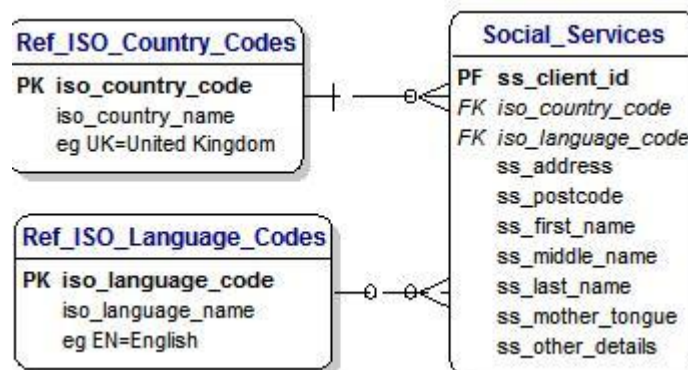
14.5 What have we learned?

MDM is becoming more important all the time. Conceptually it is quite simple, as we have seen. However, the implementation of MDM is complex and there is a wide range of products available from a number of commercial organizations.

15. Build your Own Data Models

15.1 Introduction

This chapter provides a tutorial on how to build your own data models that meet a complex set of requirements.

15.1.1 What is this?

It shows how to progress from simple models to complex models in a logical and structured manner. It starts by analyzing 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 parlor
- Making a movie

15.1.2 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.

15.1.3 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.

    - There is nothing new under the sun, and this is certainly true in the world of data modeling.

    - 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 normalized 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 modelers, data management professionals, analysts, developers and managers.

- Produce mapping specifications and plan for migration of the required data.

15.2 Opening a Pool Hall

Where do we start with our data model?

Our starting-point is the library of hundreds of data models on the Database Answers Web site:

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

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

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

Data Model for Pool League Statistics
Barry Williams
May 25th. 2009

Leagues
PK league_id
summer_or_winter
league_name
league_details

Games
PK game_code
game_name
game_description
other_details

Venues
PK venue_id
venue_name
contact_details
other_details

Teams
PK team_id
FK created_by_player_id
team_name
date_created
date_disbanded
other_details

Players
PK player_id
first_name
last_name
gender
address
other_details

Leagues_Games
PF league_id
PF game_code

Result_Possibilties
PK result_code
result_description
eg homeWin, homeLoss
eg awayWin, awayLoss

Team_Venues
PF team_id
PF venue_id
PK date_from
date_to

Team_Players
PF team_id
PF player_id
PK date_from
date_to
other_details

Matches
PK match_id
FK game_code
FK player_1_id
FK player_2_id
FK result_code
match_date
other_details

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

Our situation is that we have a fixed number of facilities, i.e. pool tables, that 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

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

Apartment_Buildings

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

Ref_Apartment_Faciltiies

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

Ref_Apartment_Types

- **PK** **apt_type_code***
- apt_type_description
- eg Studio, Duplex

Apartments

- **PK** **apt_id***
- **FK** *building_id**
- **FK** *apt_type_code*
- apt_number
- bathroom_count
- bedroom_count
- room_count
- other_apartment_details

Apartment_Facilities

- **PF** **apt_id***
- **PF** **facility_code***

Ref_Gender

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

View_Unit_Status

- **FK** *apt_id*
- **PK** **status_date***
- available_yn
- **FK** *apt_booking_id*

Apartment_Bookings

- **PK** **apt_booking_id***
- **FK** *apt_id*
- **FK** *guest_id**
- **FK** *booking_status_code**
- booking_start_date
- booking_end_date
- other_booking_details

Ref_Booking_Status

- **PK** **booking_status_code***
- booking_status_description
- eg Confirmed or Provisional

Guests

- **PK** **guest_id***
- **FK** *gender_code*
- guest_first_name
- guest_last_name
- date_of_birth
- other_guest_details

3. Driving Schools

This has Lessons that look like what we need, but it also has customer and payment details that we don't need.

Ref_Customer_Status
- (PK) customer_status_code
- customer_status_description
- eg Good, Bad.

Customers
- (PK) customer_id
- (FK) customer_address_id
- (FK) customer_status_code
- date_became_customer
- date_of_birth
- first_name
- last_name
- amount_outstanding
- email_address
- phone_number
- cell_mobile_phone_number
- other_customer_details

Staff
- (PK) staff_id
- (FK) staff_address_id
- nickname
- first_name
- middle_name
- last_name
- date_of_birth
- date_joined_staff
- date_left_staff
- other_staff_details

Addresses
- (PK) address_id
- line_1_number_building
- line_2_number_street
- line_3_area_locality
- city
- zip_postcode
- state_province_county
- country
- other_address_details

Vehicles
- (PK) vehicle_id
- vehicle_details

Ref_Lesson_Status
- (PK) lesson_status_code
- lesson_status_description
- eg Cancelled, Completed.

Lessons
- (PK) lesson_id
- (FK) customer_id
- (FK) lesson_status_code
- (FK) staff_id
- (FK) vehicle_id
- lesson_date
- lesson_time
- price
- other_lesson_details

Customer_Payments
- (PF) customer_id
- (PK) datetime_payment
- (FK) payment_method_code
- amount_payment
- other_payment_details

Ref_Payment_Methods
- (PK) payment_method_code
- payment_method_description
- eg Amex, Cash CC=Credit Card.

4. Hairdressers Appointments

Ref Staff_Job_Title
job_code
job_description

Ref Age Band
age_band_code
age_band_description

Ref Gender
gender_code
gender_description

Staff_Charges
job_code
salon_Treatment_Code
charge_amount

Staff
staff_ID
job_code
staff_Name
staff_Details
date_Joined
date_Left

Clients
client_ID
age_band_code
gender_code
client_FirstName
client_LastName
client_Address
client_AgeBand
client_CellPhone
client_HomePhone
client_EmailAddress
client_Comments
client_NaturalHairColor
client_OtherDetails

Ref Hair Style
hair_style_code
hair_style_description

Appointments
appointment_ID
client_ID
staff_ID
appointment_Date
appointment_Time
colour_used
hair_style_code
product_code

Ref Salon Treatment
salon_Treatment_Code
salon_Treatment_Description
treatment_Standard_Price

Payment_Details
payment_ID
client_ID
date_of_payment
amount_Due
amount_Paid
balance

Products
product_code
product_name
product_description
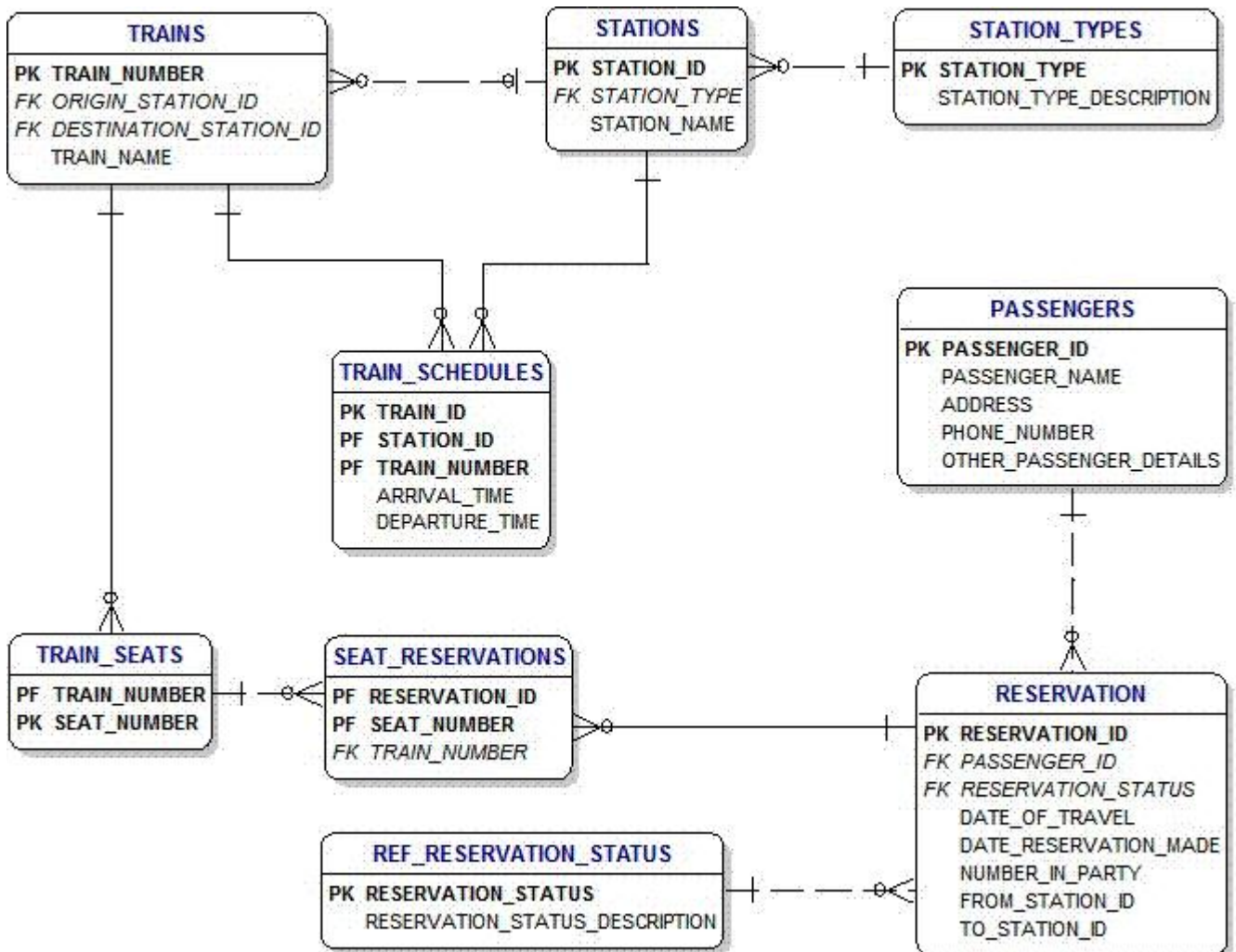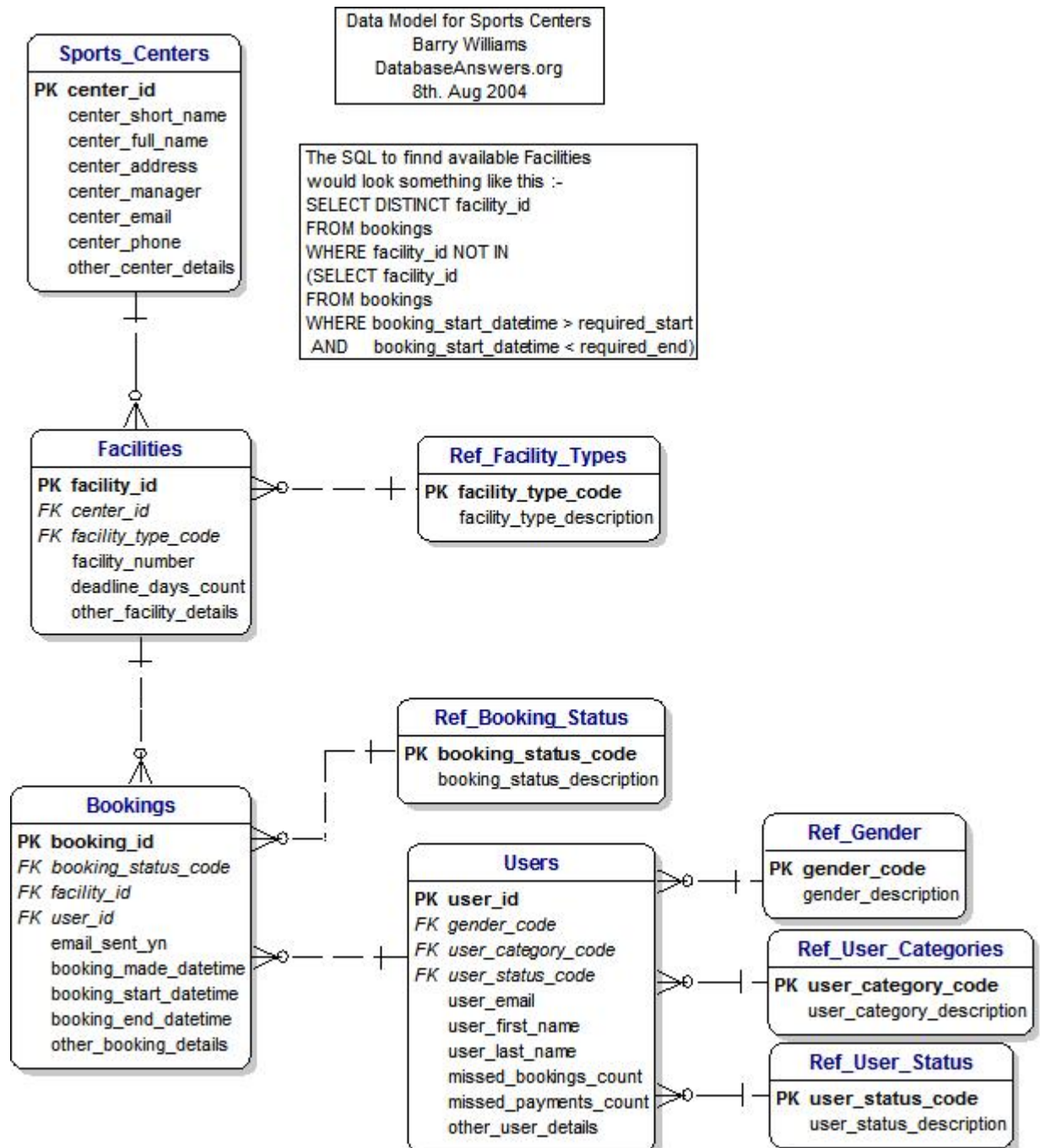other_details

6. Railway Reservations

This looks too complicated and includes things that we don't need, such as stations.

7. Sports Centers

- 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.

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:
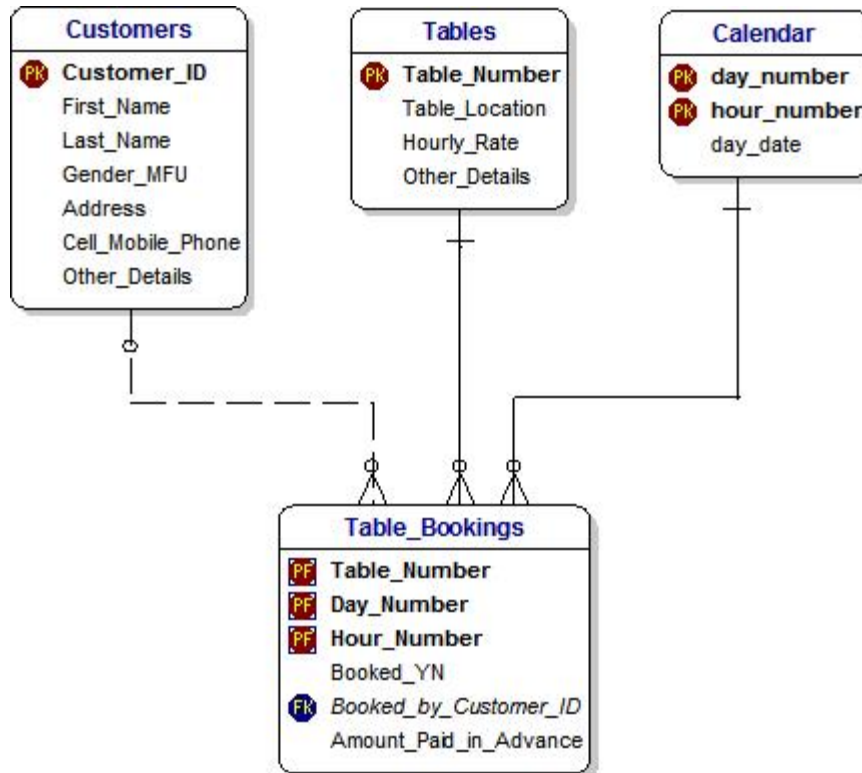
- http://www.databaseanswers.org/data_models/customers_and_credit_cards/customers_and_credit_cards_with_attributes.htm

So our data model consists of just this customer entity:
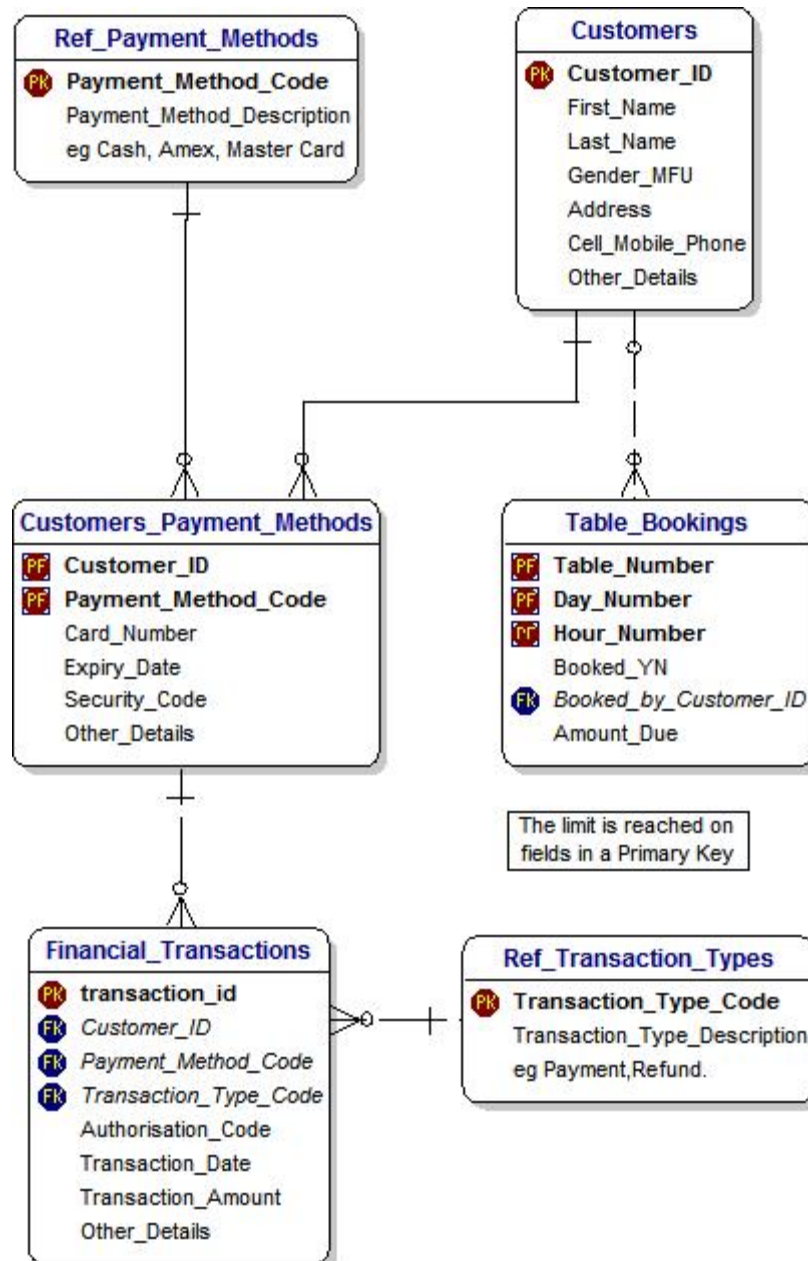
Step 3. Add Tables and Bookings

This is done in a style consistent with the Sports Centers model. Full payment is made in advance.

Step 4. Add Customers and Payment Methods

Then we add payments 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 6. 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.
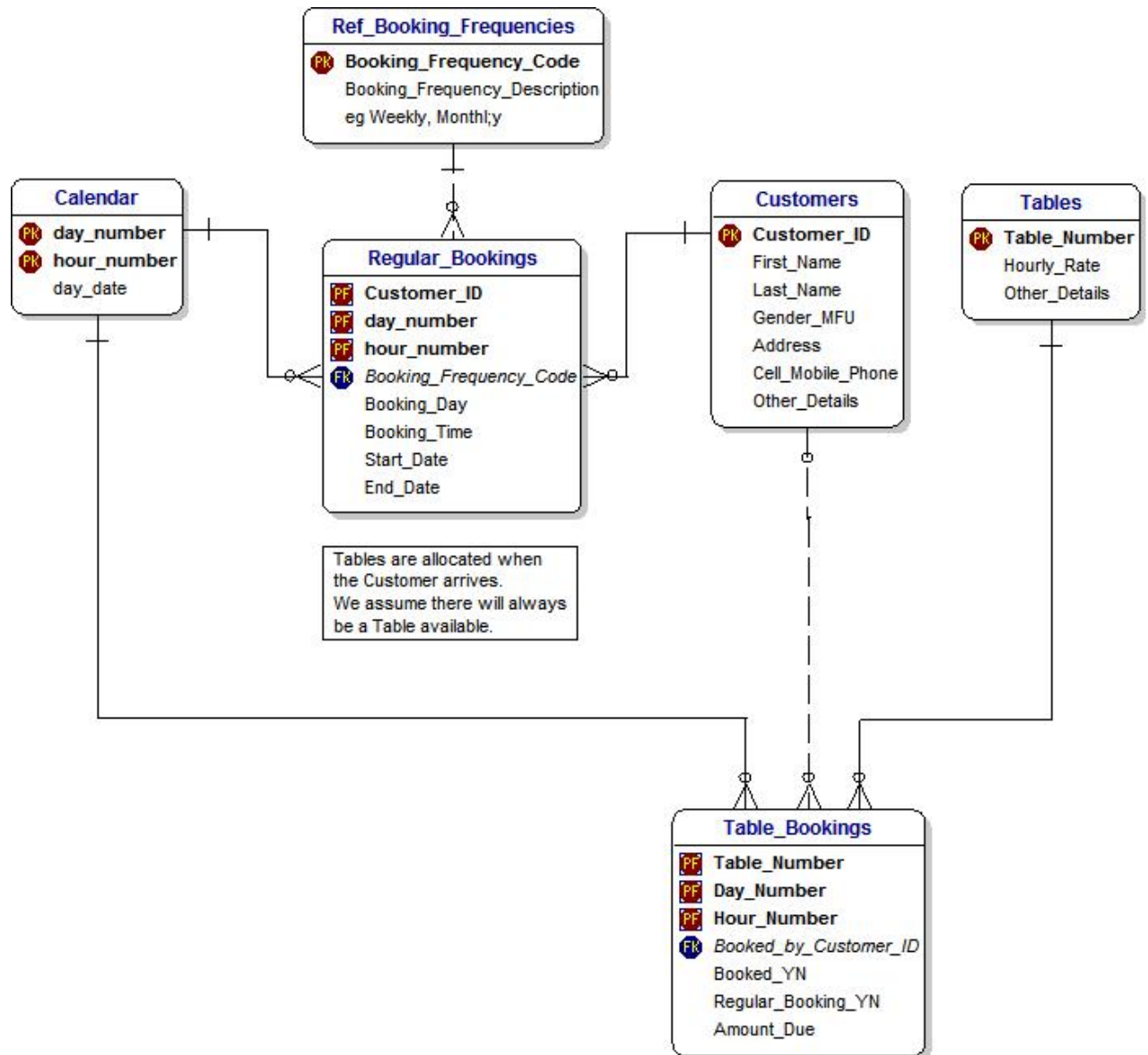
Step 7. Add Inheritance

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

Generic_Customers

- (PK) Customer_ID
- First_Name
- Last_Name
- Address
- Gender_MFU
- Cell_Mobile_Phone
- Other_Details

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

Personal_Customers

- (PF) Customer_ID
- SSN_or_NI_Number
- Other_Personal_Details

Commercial_Customers

- (PF) Customer_ID
- UK_VAT_Number
- Other_Details

Step 8. 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.
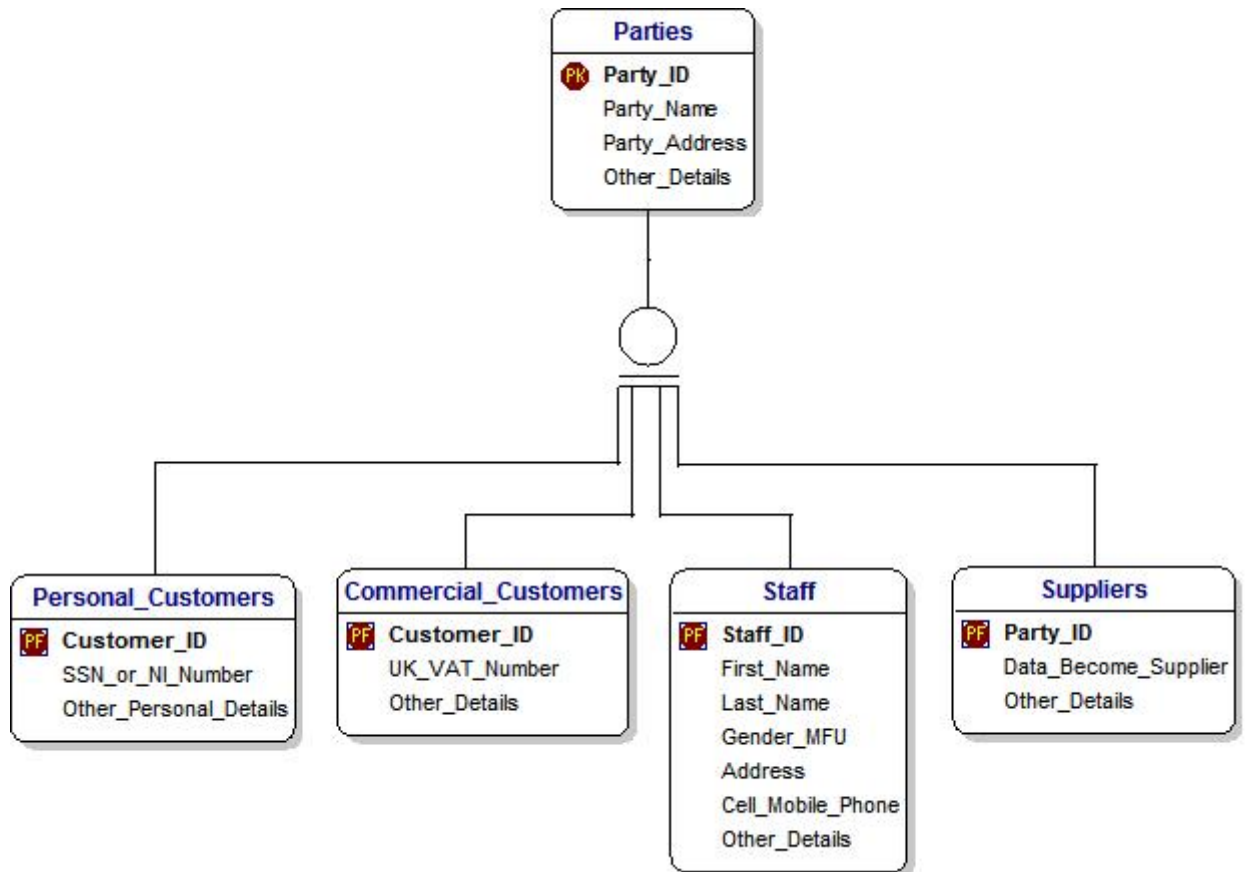
Step 9. Reference Data

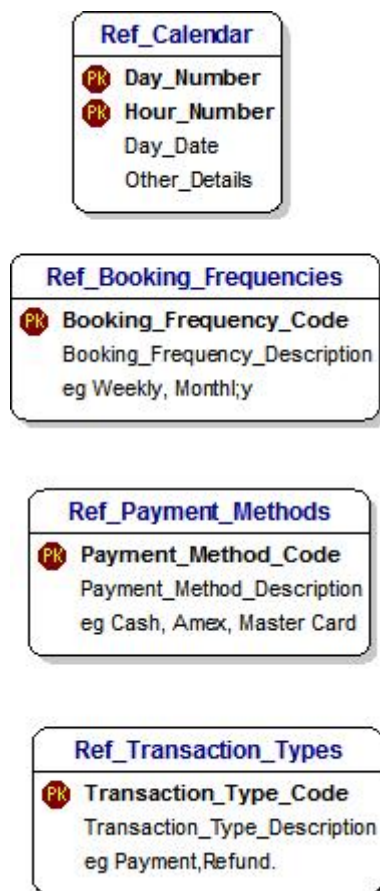This shows all of the reference data that we have referred to so far in our pool hall management data model.

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.

Ref_Calendar

PK	Day_Number
PK	Hour_Number
	Day_Date
	Other_Details

Ref_Booking_Frequencies

PK	Booking_Frequency_Code
	Booking_Frequency_Description
	eg Weekly, Monthl;y

Ref_Payment_Methods

PK	Payment_Method_Code
	Payment_Method_Description
	eg Cash, Amex, Master Card

Ref_Transaction_Types

PK	Transaction_Type_Code
	Transaction_Type_Description
	eg Payment,Refund.

Step 10. 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 modelers who want to see all the details of every table.
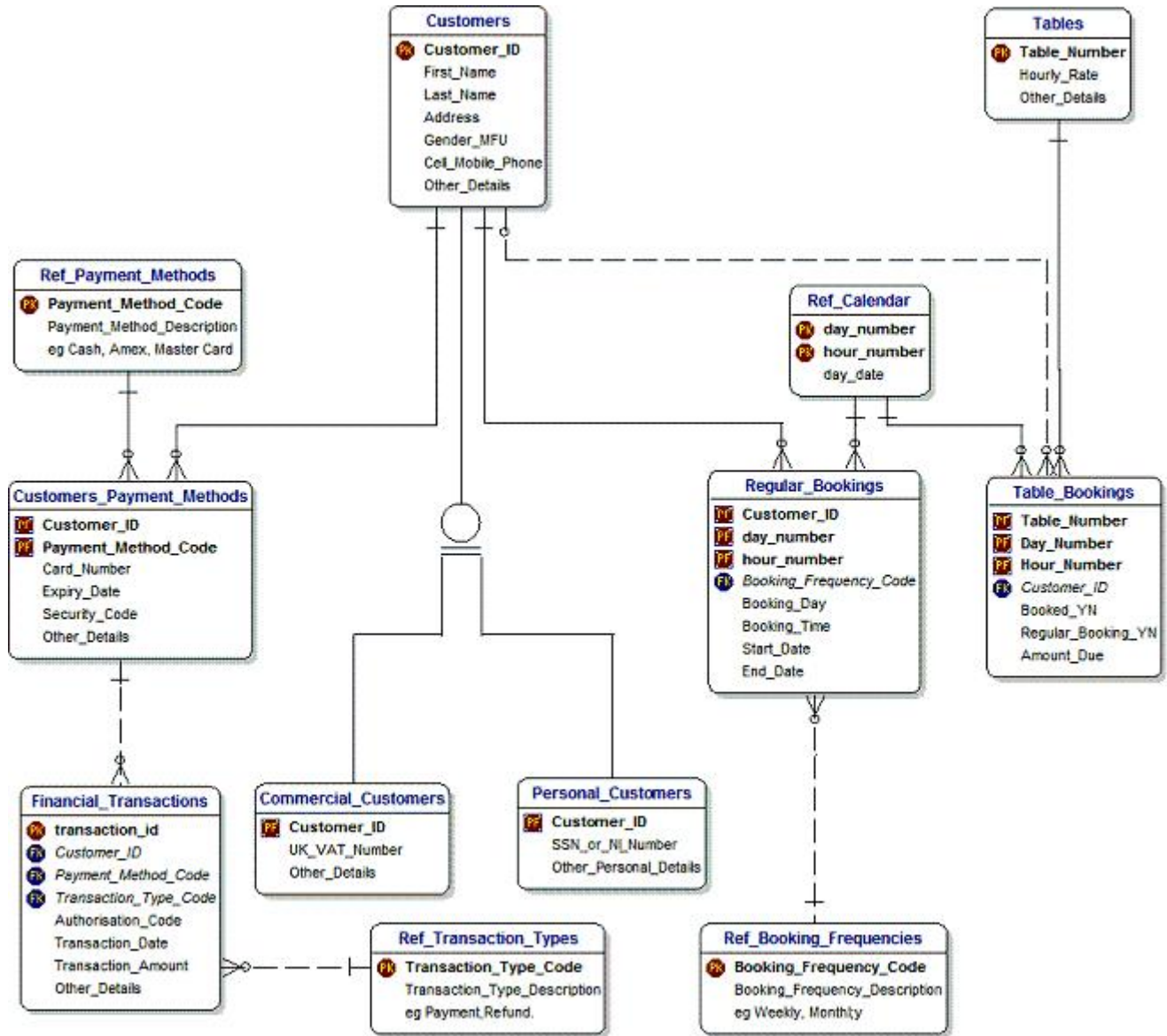
Step 13. The Complete Model - Showing Key Fields Only

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 modelers.
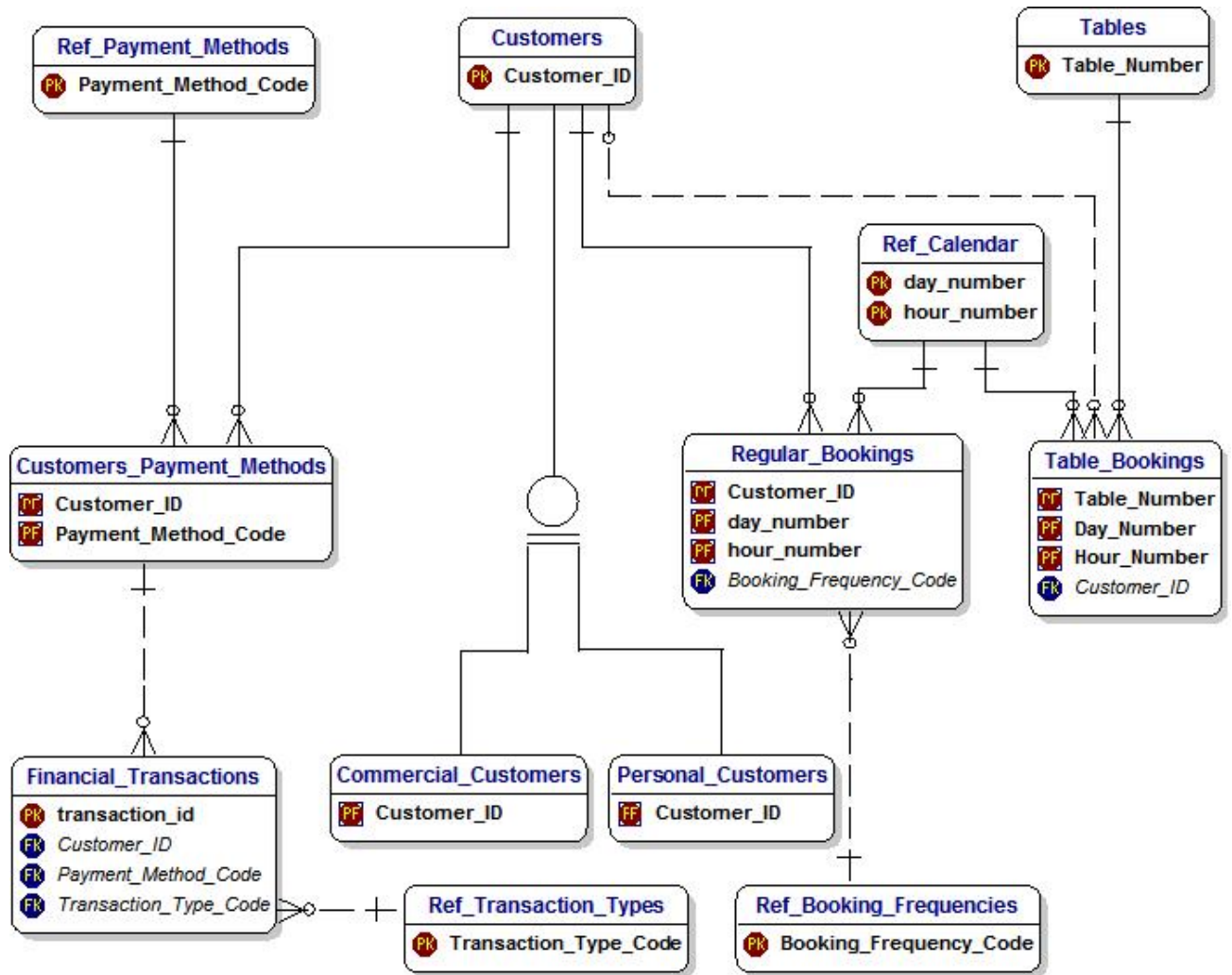
Step 14. The Complete Model - Showing Entity Names Only

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.
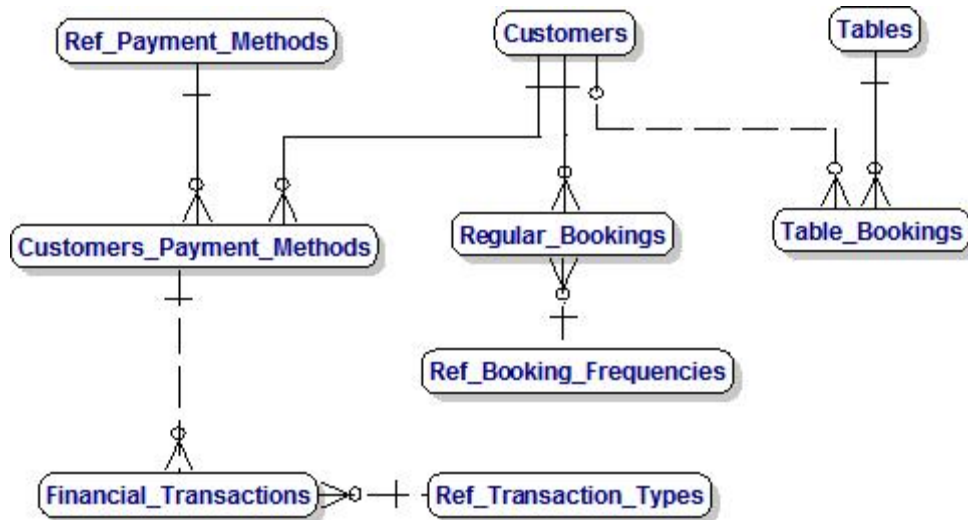
Step 11. The Complete Model - Showing Entity Names Only Without Reference Data Tables

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 do not have to.

- Regular bookings must always be associated with a customer.
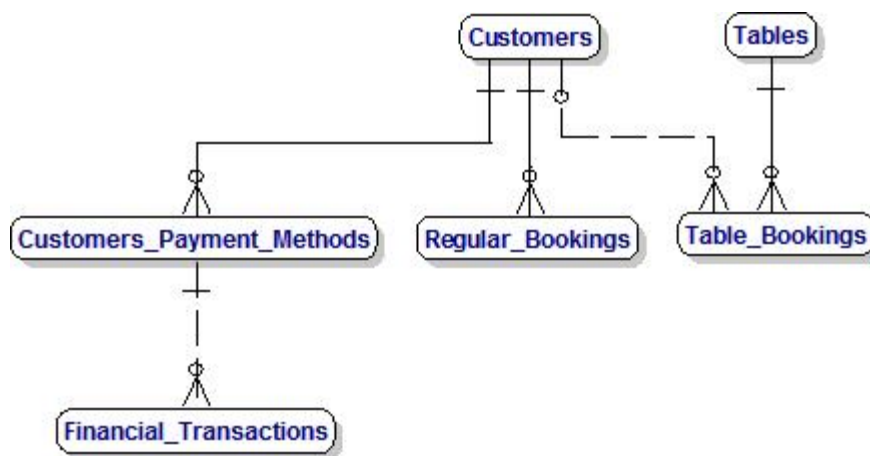
Step 15. Top-Level and Subject Area Models

The subject areas include:

- Bookings

- Customers

- Payments

- Data Warehouse

12.3.3.1 Top-Level Model

The Top-Level Model goes here.

12.3.3.2 Subject Area Models

12.3.3.3 Bookings

The Bookings Model goes here.

12.3.3.4 Customers

The Customers Model goes here.

12.3.3.5 Payments

The Payments Model goes here.

12.3.3.6 Data Warehouse

The Data Warehouse Model goes here.

15.3 Opening a Funeral Home

Here is another finished product:

- 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*.
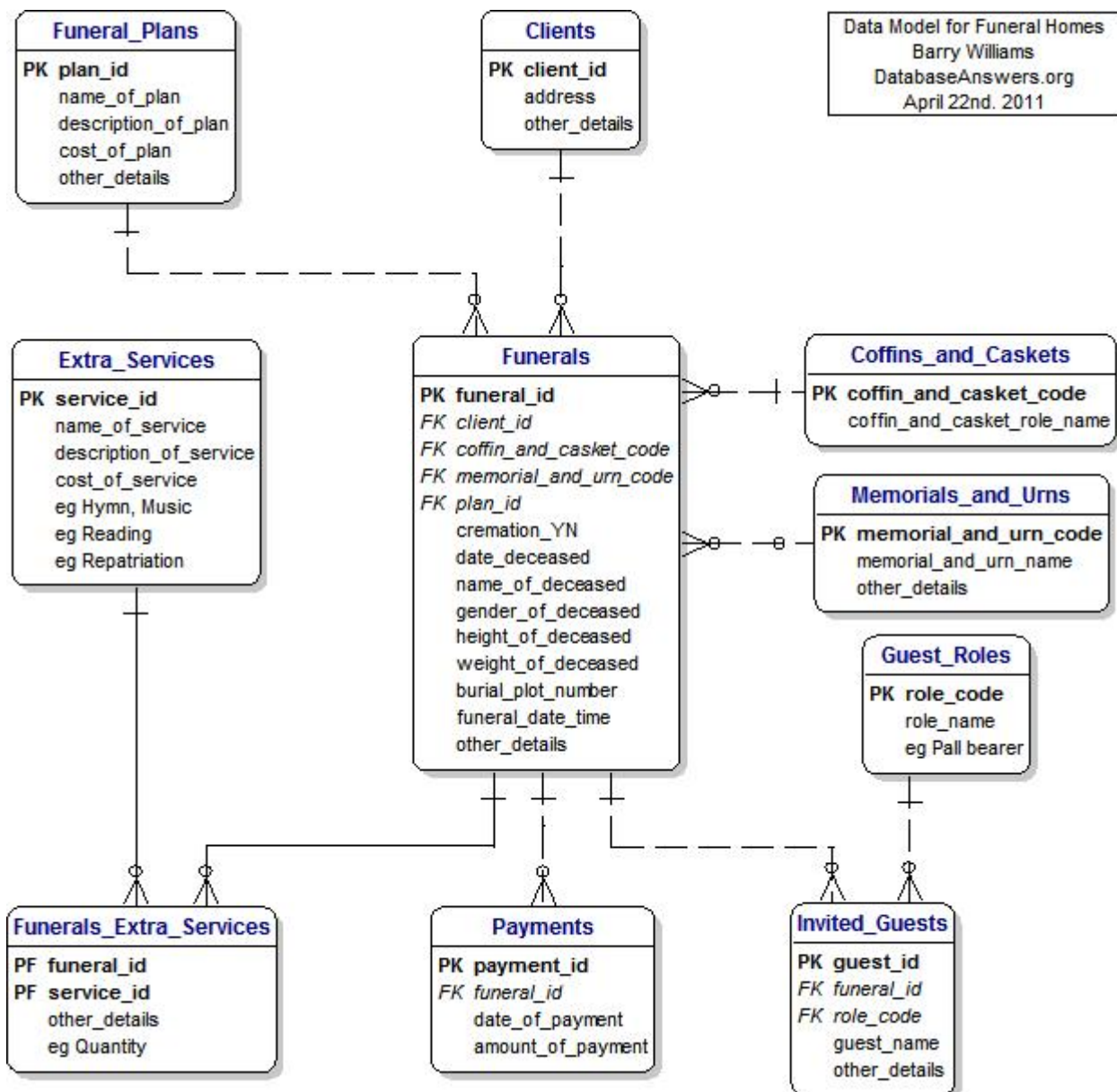
Funeral_Plans

PK plan_id
- name_of_plan
- description_of_plan
- cost_of_plan
- other_details

Clients

PK client_id
- address
- other_details

Data Model for Funeral Homes
Barry Williams
DatabaseAnswers.org
April 22nd. 2011

Extra_Services

PK service_id
- name_of_service
- description_of_service
- cost_of_service
- eg Hymn, Music
- eg Reading
- eg Repatriation

Funerals

PK funeral_id
FK client_id
FK coffin_and_casket_code
FK memorial_and_urn_code
FK plan_id
- cremation_YN
- date_deceased
- name_of_deceased
- gender_of_deceased
- height_of_deceased
- weight_of_deceased
- burial_plot_number
- funeral_date_time
- other_details

Coffins_and_Caskets

PK coffin_and_casket_code
- coffin_and_casket_role_name

Memorials_and_Urns

PK memorial_and_urn_code
- memorial_and_urn_name
- other_details

Guest_Roles

PK role_code
- role_name
- eg Pall bearer

Funerals_Extra_Services

PF funeral_id
PF service_id
- other_details
- eg Quantity

Payments

PK payment_id
FK funeral_id
- date_of_payment
- amount_of_payment

Invited_Guests

PK guest_id
FK funeral_id
FK role_code
- guest_name
- other_details

15.4 Making a Movie

15.4.1 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

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.
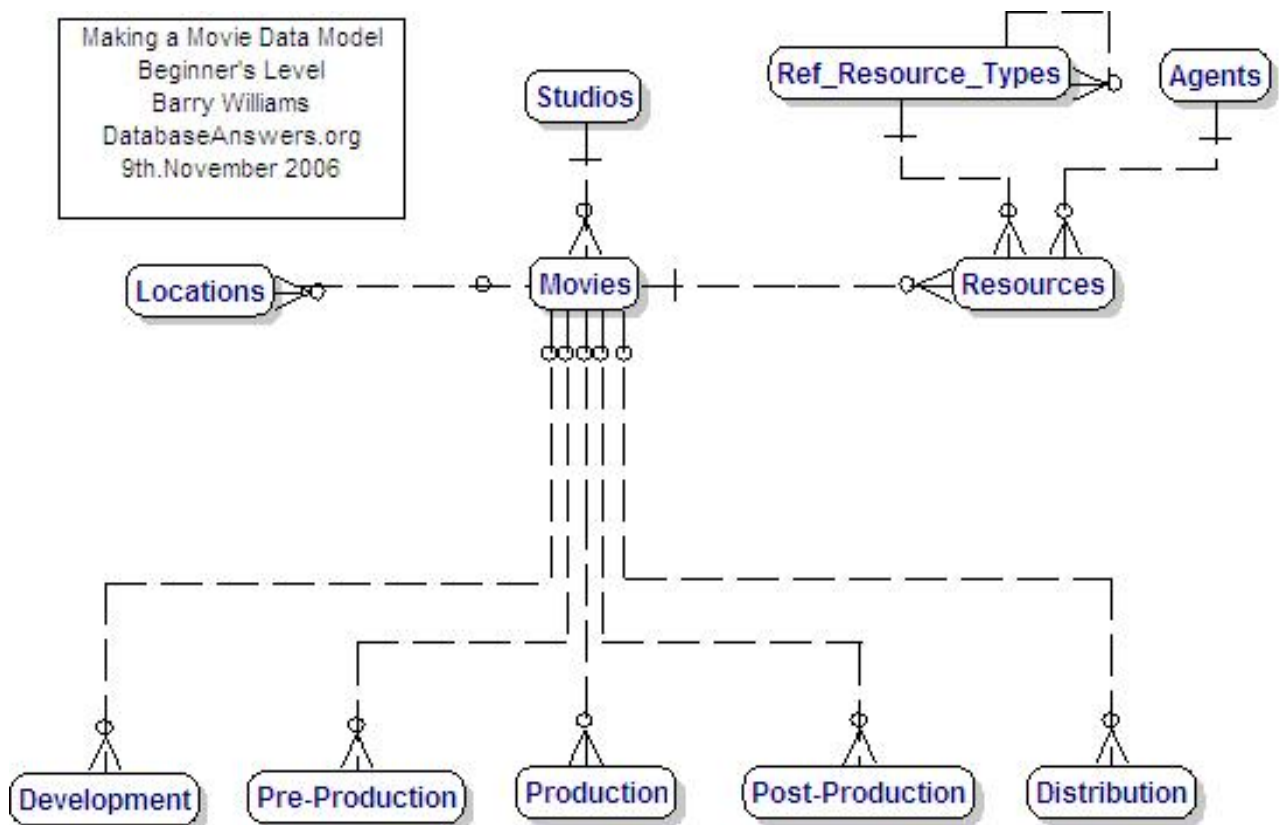
Making a Movie Data Model
Beginner's Level
Barry Williams
DatabaseAnswers.org
9th.November 2006

Studios

Ref_Resource_Types   **Agents**

Locations   **Movies**   **Resources**

Development   **Pre-Production**   **Production**   **Post-Production**   **Distribution**

15.4.2 A Complex Model

Here's a complex version:

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

This shows a different approach to modeling 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.
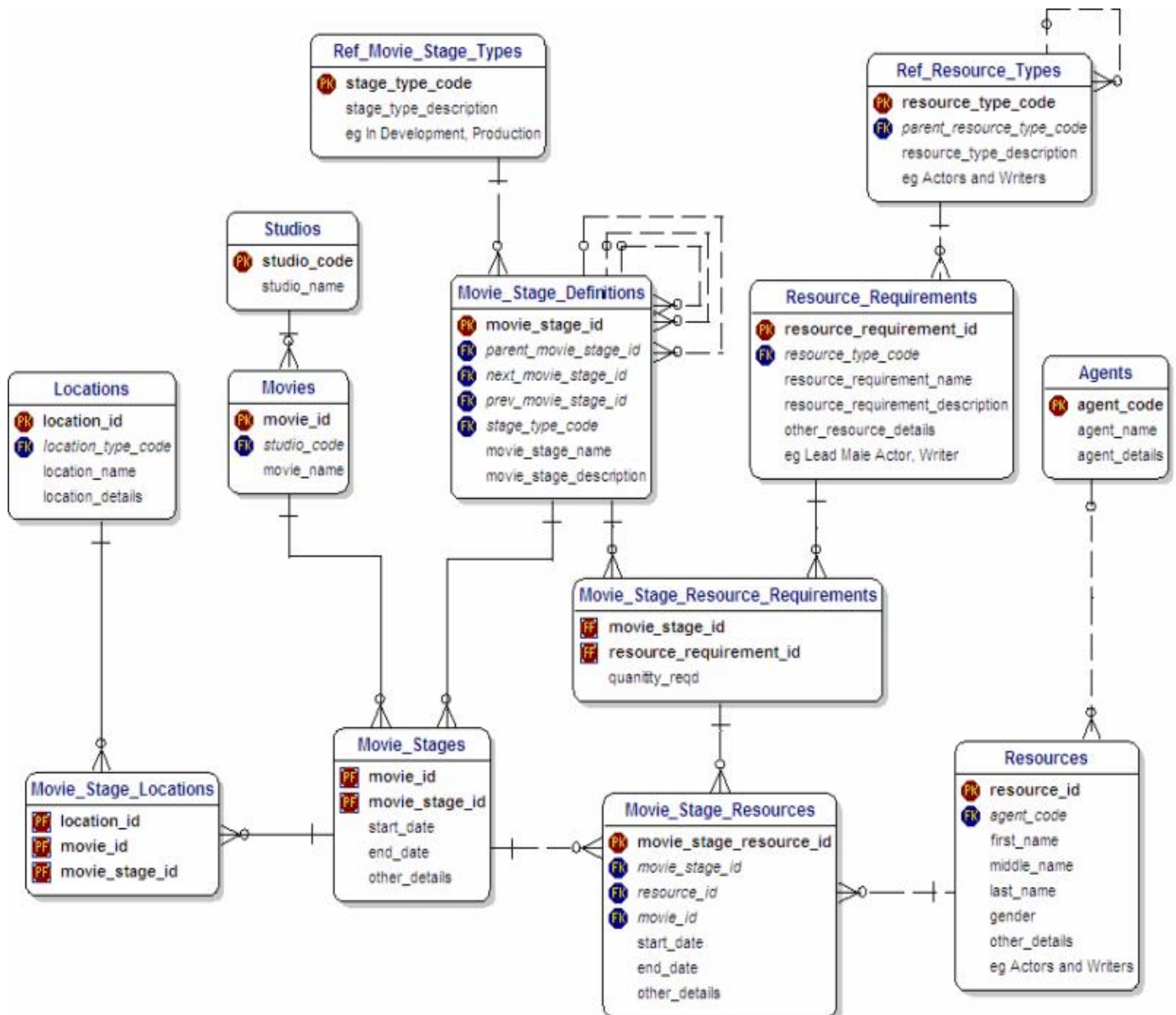
15.5 What have we learned?

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 modeling in your professional life because that situation is very common.

You will therefore emerge looking like a star ;0)

16. Establishing a Single View of the Truth

16.1 Introduction

16.1.1 What is this?

A 'single view of the truth' means that any large enterprise is able to identify uniquely any reference to customers, products and so on.

A simple example of this problem is where residents in a UK local authority are identified with variations of their names so they are sometimes not recognized as the same person.

For example, in the UK, Joe Bloggs, Joey Bloggs and Joesph Bloggs and Mr. J. Bloggs might all be the same person but unless special software is used to match these alternatives to what are essentially the same name, then Joe will not be recognized as who he is and the service he is offered will not reflect a true understanding of his situation.

In the US, the equivalent would be John Doe, Johnny Doe and so on.

16.1.2 Why is it important?

We are currently experiencing a global recession.

This makes it imperative that major commercial enterprises are able to obtain trustworthy figures in their business intelligence and performance reports. This is also a legal requirement thanks to the Sarbanes-Oxley Act.

The material in this chapter is amplified in these two data models on the Database Answers Web site:

- Master Data Management:
  http://www.databaseanswers.org/data_models/master_data_mgt/index.htm

- Product Catalogues:
  http://www.databaseanswers.org/data_models/product_catalogs/index.htm

16.1.3 What Will I Learn?

You will learn how to quickly recognize and understand situations where it is important to establish a 'single view of the truth'.

You will also learn the techniques of data management that make this possible and the underlying data models.

It is rather surprising that the problem seems complex but the techniques are relatively simple.

16.2 The Data Architecture

We start with a simple four-layer **data architecture**.

This covers the activities from data sources (CRM, order processing and billing), through the consolidated data layer, where the single view of the 'things of interest' comes in, up to the generic data mart.
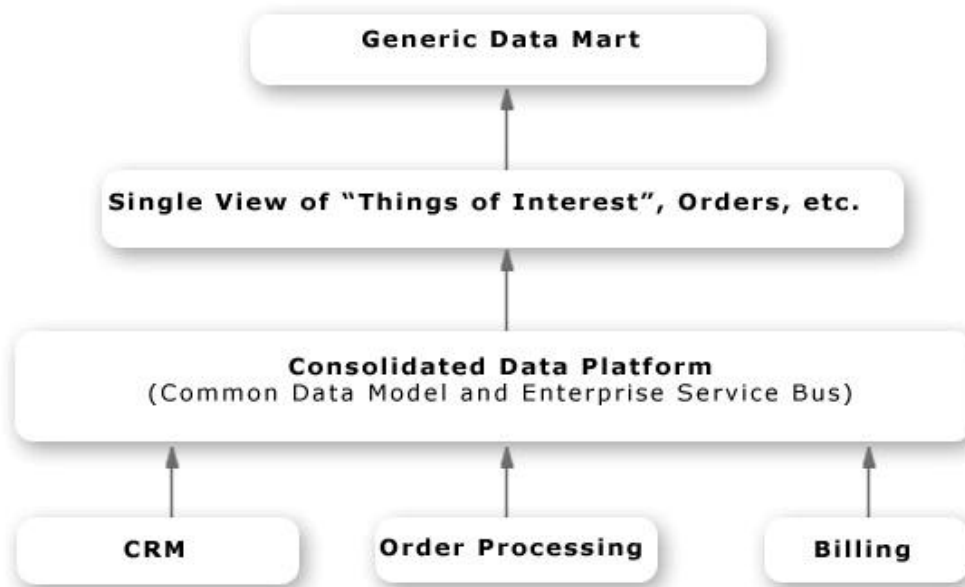
16.3 A Data Loading Sequence

This diagram shows the sequence in which data has to be loaded because of mutual dependencies.

16.4 Detailed Data Architecture

This e Data Architecture consists of four data layers, as shown in this diagram.

```
┌─────────────────────────────────┐        ┌──────────────────────┐
│   BI + Financial Reports Layer  │        │ Sustainable          │
└─────────────────────────────────┘        │ Semantic             │
                 ▲                          │ Layer      (BI       │
┌─────────────────────────────────┐        │ Platform on          │
│        Data Marts Layer         │        │ page 6 of EBIE       │
└─────────────────────────────────┘        │ Ref Arch)            │
                 ▲                          └──────────────────────┘
┌─────────────────────────────────┐
│    Enterprise Data Warehouse    │        ┌──────────────────────┐
└─────────────────────────────────┘        │ SEDW                 │
                 ▲                          └──────────────────────┘
┌─────────────────────────────────┐
│     Mapping from CDM to EDW     │
└─────────────────────────────────┘
                 ▲
┌─────────────────────────────────┐
│ Integration Data Platform    -  │
│ Single Version of the Truth  -  │
│ MDM            - Canonical       │        ┌──────────────────────┐
│ Data Model (CDM)    - Staging    │        │ Staging Area         │
│ Area / ETL                       │        └──────────────────────┘
└─────────────────────────────────┘
                 ▲
┌─────────────────────────────────┐
│ Financial and ODS Layer    -    │
│ EDW 1.0, FACT, (GCSS,RKEM, etc.)│
└─────────────────────────────────┘
┌──────────────────────┐    ▲
│ Container Lifecycle … │
└──────────────────────┘
```
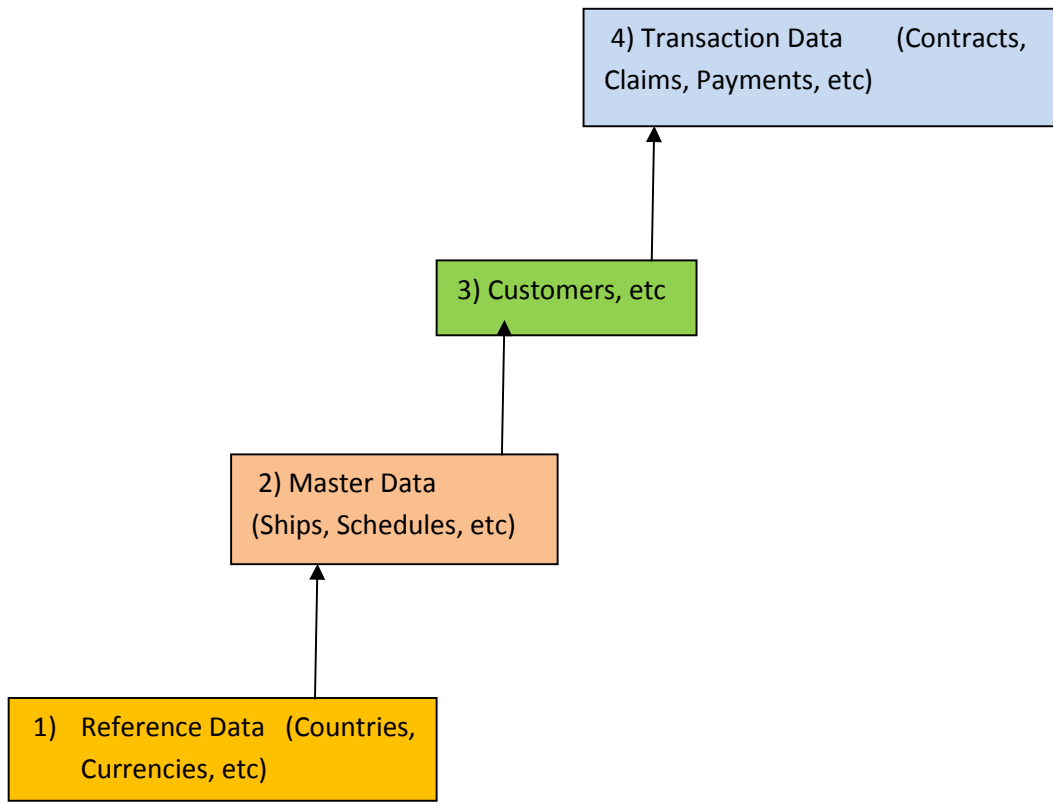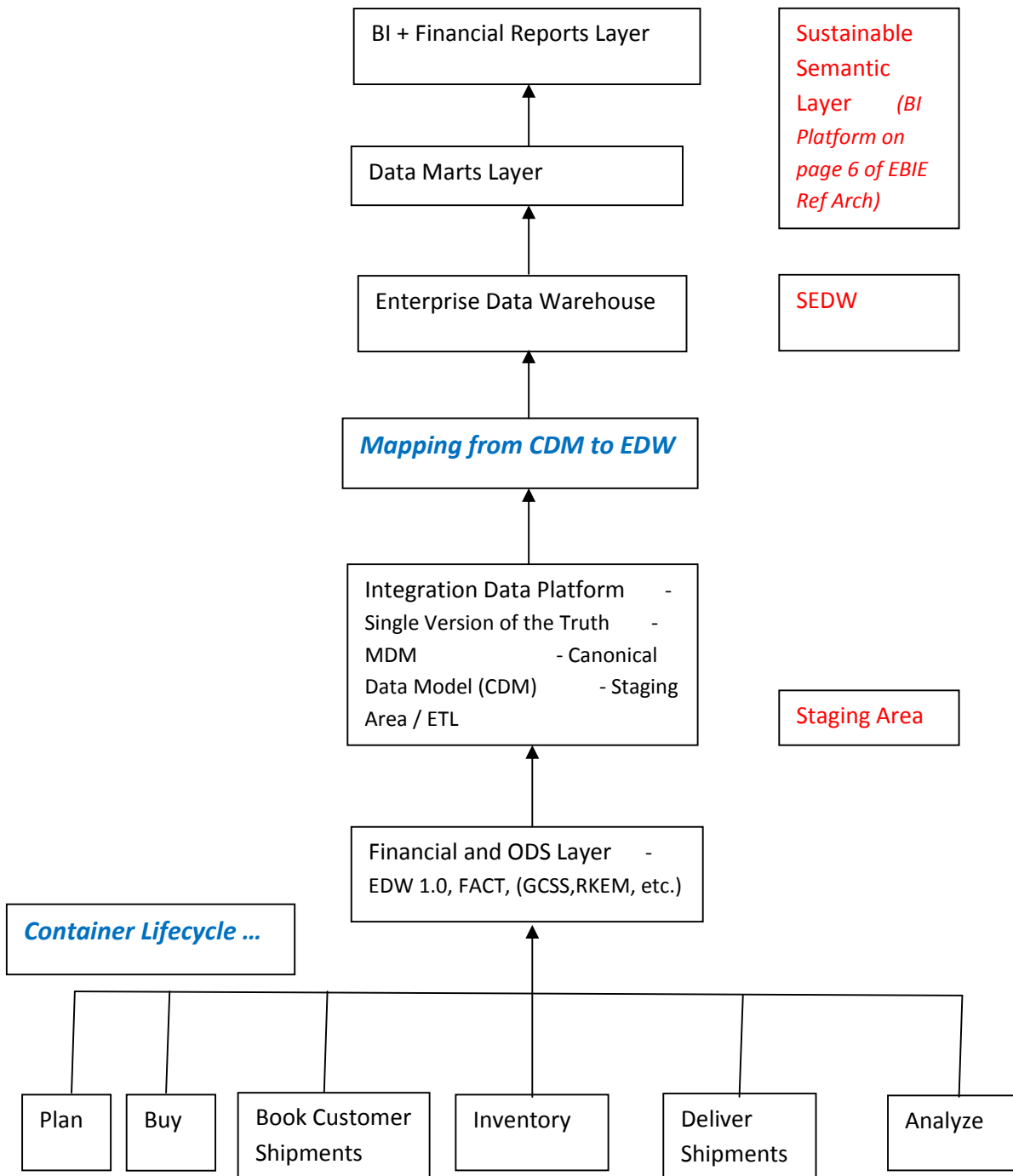
Plan	Buy	Book Customer Shipments	Inventory	Deliver Shipments	Analyze

16.5 The Canonical Data Model

The **canonical** data model is the smallest model that can be designed that provides useful functionality and can be used for:

- Establishing communication with business users
- Transforming data to for loading into data warehouse
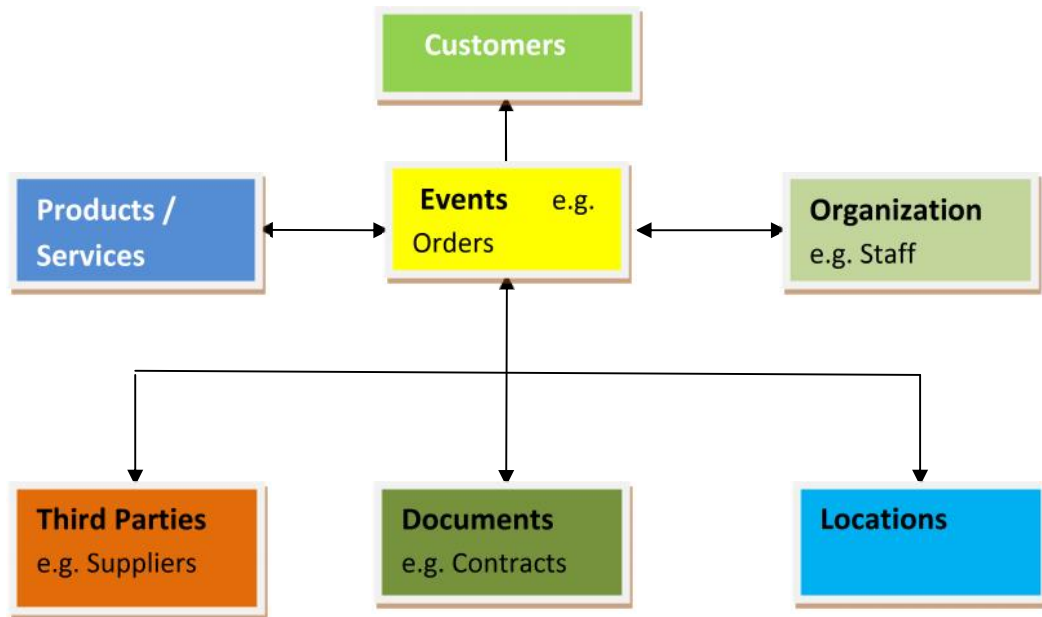
16.6 ERD for the Canonical Data Model

This shows the starting point, which is the generic model for the EDM.

Business Areas
- Business_Area_ID
- Business_Area_Name
- Business_Area_Description

Customers
- Customer_ID
- Address_ID (FK)
- Customer_Type_Code (FK)
- TK_Maxx_Care_YNU
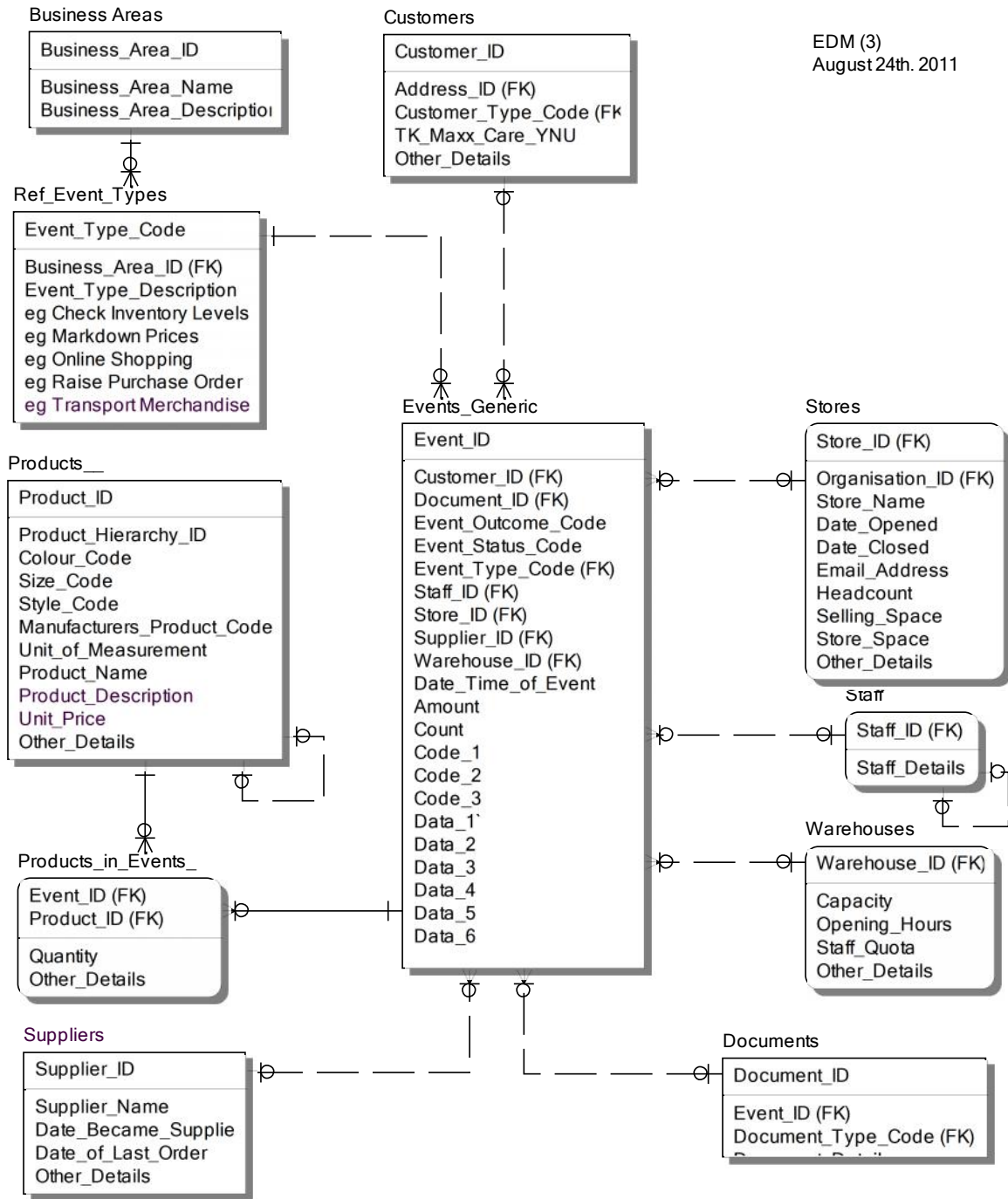- Other_Details

EDM (3)
August 24th. 2011

Ref_Event_Types
- Event_Type_Code
- Business_Area_ID (FK)
- Event_Type_Description
- eg Check Inventory Levels
- eg Markdown Prices
- eg Online Shopping
- eg Raise Purchase Order
- eg Transport Merchandise

Products__
- Product_ID
- Product_Hierarchy_ID
- Colour_Code
- Size_Code
- Style_Code
- Manufacturers_Product_Code
- Unit_of_Measurement
- Product_Name
- Product_Description
- Unit_Price
- Other_Details

Events_Generic
- Event_ID
- Customer_ID (FK)
- Document_ID (FK)
- Event_Outcome_Code
- Event_Status_Code
- Event_Type_Code (FK)
- Staff_ID (FK)
- Store_ID (FK)
- Supplier_ID (FK)
- Warehouse_ID (FK)
- Date_Time_of_Event
- Amount
- Count
- Code_1
- Code_2
- Code_3
- Data_1`
- Data_2
- Data_3
- Data_4
- Data_5
- Data_6

Stores
- Store_ID (FK)
- Organisation_ID (FK)
- Store_Name
- Date_Opened
- Date_Closed
- Email_Address
- Headcount
- Selling_Space
- Store_Space
- Other_Details

Staff
- Staff_ID (FK)
- Staff_Details

Warehouses
- Warehouse_ID (FK)
- Capacity
- Opening_Hours
- Staff_Quota
- Other_Details

Products_in_Events_
- Event_ID (FK)
- Product_ID (FK)
- Quantity
- Other_Details

Suppliers
- Supplier_ID
- Supplier_Name
- Date_Became_Supplie
- Date_of_Last_Order
- Other_Details

Documents
- Document_ID
- Event_ID (FK)
- Document_Type_Code (FK)
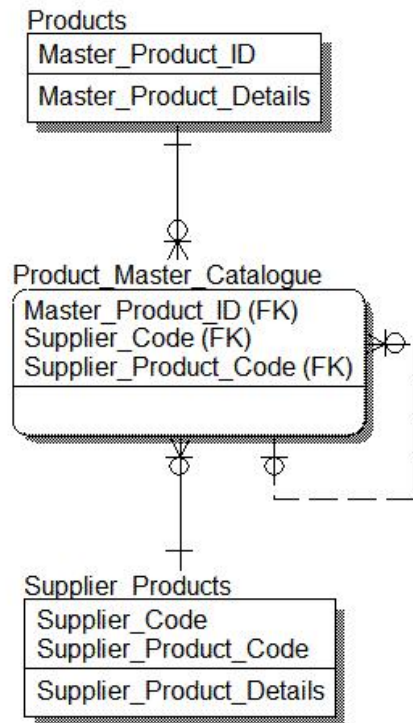
16.7 Master Data Management

16.7.1 Customer Master Index

This shows a **customer master index** in a local authority environment.
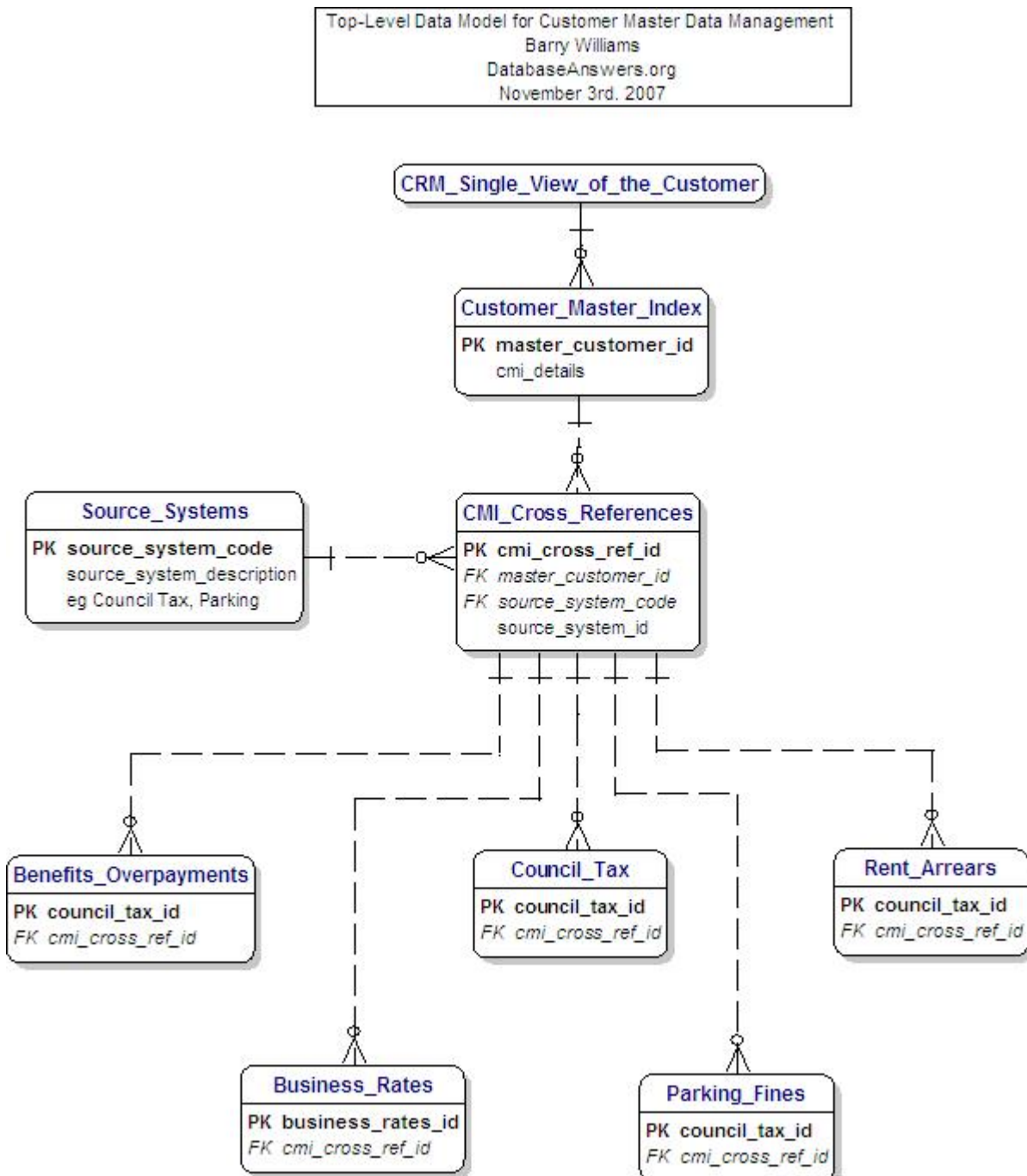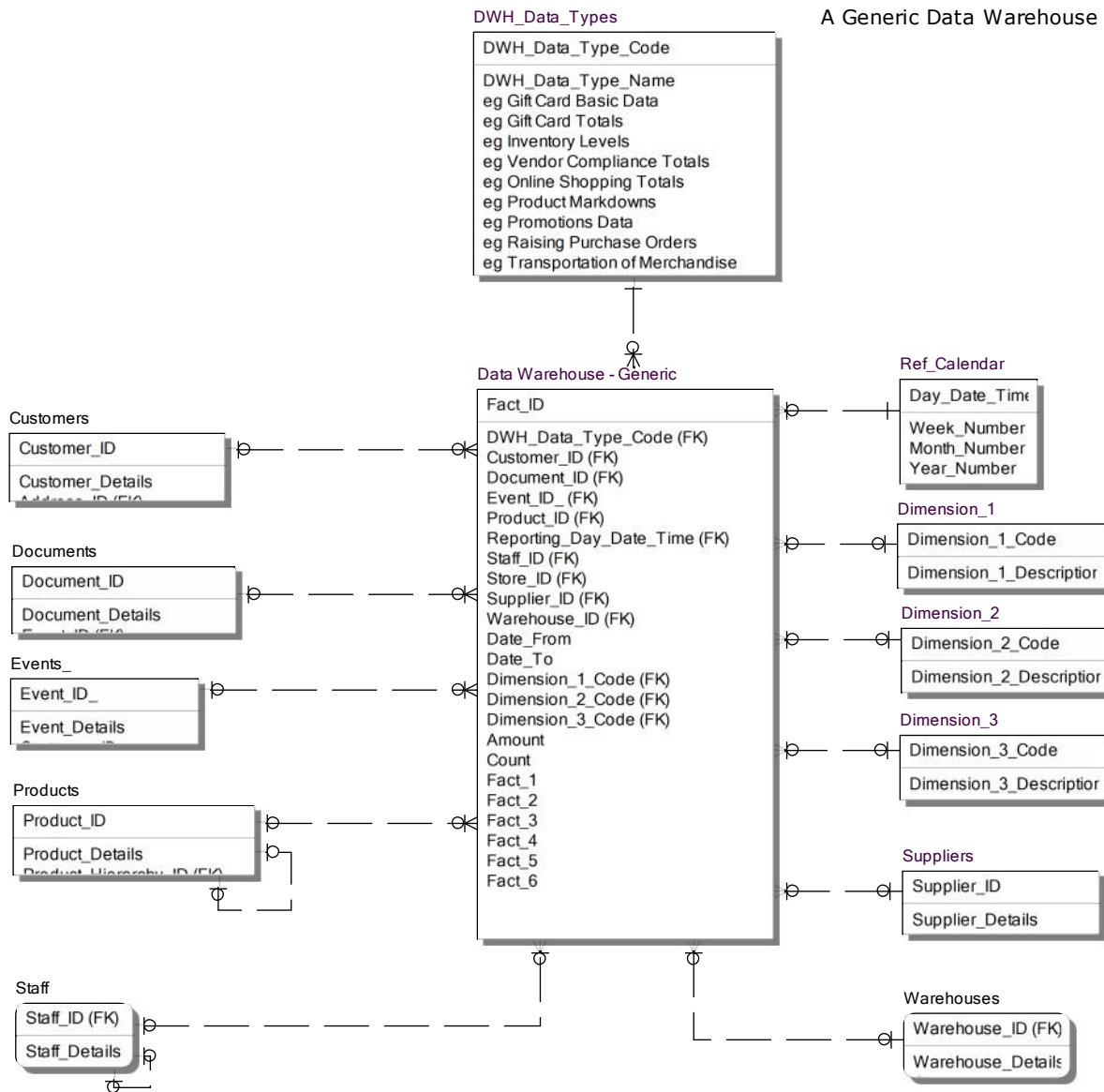
16.7.2 Product Master Catalogue

This data model shows that the **product master catalogue** plays a central role in matching specific supplier product codes to one unique master product ID.

Products and services and related hierarchies are a very important part of any enterprise data landscape.

At the logical level, a product or service hierarchy can be defined in a concise manner that offers a high degree of flexibility. This design makes it possible to establish multiple lower level hierarchy chains that belong to the same chain at a higher level. In other words, to have varying degrees of depth in the hierarchy for different products or services.

16.8 ERD for Generic Data Warehouse

DWH_Data_Types

DWH_Data_Type_Code

DWH_Data_Type_Name
eg Gift Card Basic Data
eg Gift Card Totals
eg Inventory Levels
eg Vendor Compliance Totals
eg Online Shopping Totals
eg Product Markdowns
eg Promotions Data
eg Raising Purchase Orders
eg Transportation of Merchandise

A Generic Data Warehouse

Data Warehouse - Generic

Fact_ID

DWH_Data_Type_Code (FK)
Customer_ID (FK)
Document_ID (FK)
Event_ID_ (FK)
Product_ID (FK)
Reporting_Day_Date_Time (FK)
Staff_ID (FK)
Store_ID (FK)
Supplier_ID (FK)
Warehouse_ID (FK)
Date_From
Date_To
Dimension_1_Code (FK)
Dimension_2_Code (FK)
Dimension_3_Code (FK)
Amount
Count
Fact_1
Fact_2
Fact_3
Fact_4
Fact_5
Fact_6

Customers

Customer_ID

Customer_Details
Address_ID (FK)

Documents

Document_ID

Document_Details
Event_ID (FK)

Events_

Event_ID_

Event_Details
Customer_ID

Products

Product_ID

Product_Details
Product_Hierarchy_ID (FK)

Staff

Staff_ID (FK)

Staff_Details

Ref_Calendar

Day_Date_Time

Week_Number
Month_Number
Year_Number

Dimension_1

Dimension_1_Code

Dimension_1_Description

Dimension_2

Dimension_2_Code

Dimension_2_Description

Dimension_3

Dimension_3_Code

Dimension_3_Description

Suppliers

Supplier_ID

Supplier_Details

Warehouses

Warehouse_ID (FK)

Warehouse_Details

16.9 Generic Retail Data Mart

This design shows two date fields because that is a common pattern. In a similar way, it shows six dimensions and six facts for illustrative purposes but these could be any number of attributes.
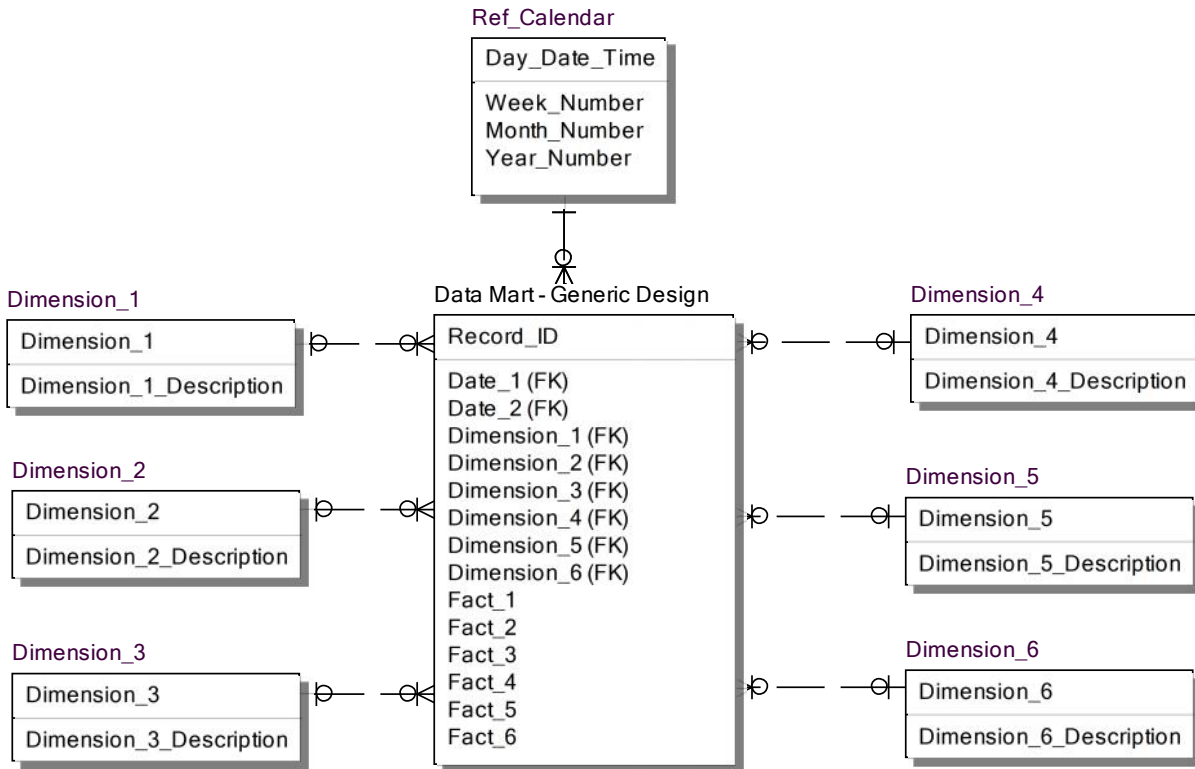
16.10 Sample Data Marts

This shows the design for three representative online shopping data marts.

16.11 What have we learned?

We have learned why a 'single view of the truth' is very important and why it is often difficult to obtain in a large commercial enterprise.

Appendix A. Checklist for Quality Assurance

This Checklist is based on the concept of the data model scorecard that was originated by Steve Hoberman.

It is discussed in Hoberman's latest book, *Data Modeling for the Business*.

Nr	FEATURE	IMPORTANCE	YES/NO	COMMENT
1	Can the scope of the model be defined?	Essential		There must be c with the busines
2	Can the requirements be defined?	Essential		Requirements m
3	Does the model meet the requirements?	Essential		The model must
4	Is there a comprehensive glossary of terms?	Essential		Very important recognized.
5	Have comprehensive business rules been defined?	Essential		Very important recognized.
6	Normalization checks (good value of Y or N) 1. Can the values of every data item in a table be derived only from the primary key? (Y) 3. Can any data item be derived from other items? (N) 4. Do any column names repeat in the same table? (N) 6. Does the same item of data appear in more than one table? (N)	Desirable		The model migh these checks. The power and f approach makes errors and still p operational data
7	Is there compliance with any relevant data standards?	Desirable		Not essential.
8	Does the model use relevant design patterns?	Desirable		Might be OK eve identified. But if it isn't laic the professional model.
9	Is the model easy to read and understand?	Desirable		The flow of logic top-left to botto It might be OK e understood. But if it isn't laic the professional

				model.
12	Can any repeating groups be identified?	Not critical		Not critical beca Usually indicate performance rea be acceptable fo For example, ad
12	Can any derived data be identified?	Not critical		Not critical beca
13	Does the model follow naming standards?	Not critical		Cosmetic but is professionalism.
---	---	SUMMARY		

If the answer to all the essential features is 'Yes' then the model is acceptable.

If any of the essential questions have a 'No' answer, the model is not acceptable.

Any 'No' answers to 'Desirable' or 'Not critical' questions do not affect the acceptability of the model but mean that it could be improved.

RESULT	RATING	COMMENT
All essential features are Yes	Acceptable	
Any essential features are No	Not Acceptable	The essential items are top priority for improvement.

Typical Summary
The results of a typical model might result in this summary:

"Reservations are that the documentation does not demonstrate that the data model meets the user requirements. The data model shows some weaknesses that the supplier has agreed to address."

Follow-Up Remedial Action
A reasonable result of a QA analysis would be the identification of some problems that could be rectified fairly easily and quickly. This applies to things like documentation and naming standards. The appropriate remedial action will depend on the context and scope of the data model.

For a Health Check:

No action is required beyond the presentation of a report because the QA is simply to establish the 'as-is' situation.

For a proposed Application:

It is essential that the model accurately meets the user requirements. If it does not, then it must be corrected in discussion with the users and the modeler.

For Data Migration:

It is essential that the model is correct at the detailed level of tables, fields and data types.
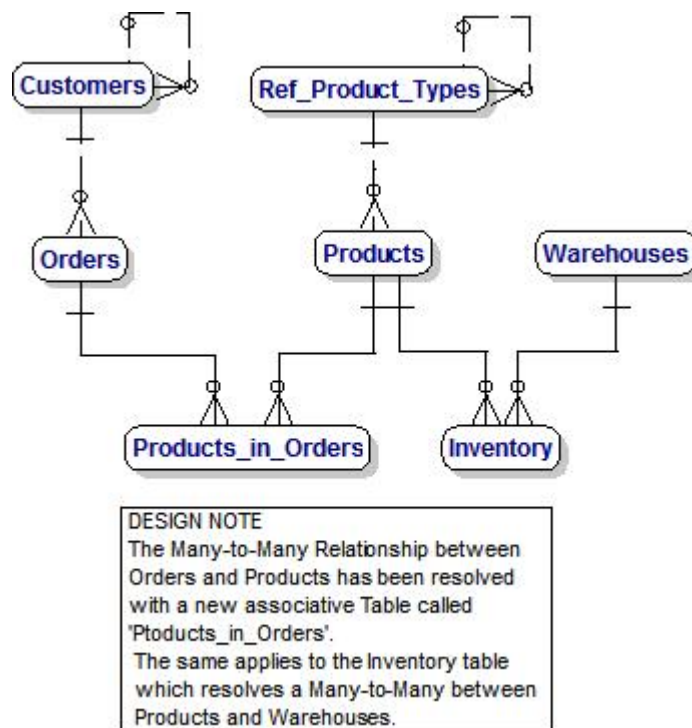
Appendix B. A Case Study

This case study provides an example of the tutorial in action. It includes blank templates and sample templates that are guidelines.

Step 1. Create a Top-Level Business Data Model

1.1 Background

Let's assume that a data model has been provided by a third party. The first step is to understand the data model by creating a top-level business data model.

Here is the data model that we will use as an example:

DESIGN NOTE
The Many-to-Many Relationship between Orders and Products has been resolved with a new associative Table called 'Ptoducts_in_Orders'.
The same applies to the Inventory table which resolves a Many-to-Many between Products and Warehouses.

1.2 Our Conclusions

Our conclusions are that this is not a good data model.

Reasons include:

- It contains reference data that is not appropriate at the top level.
- There is no description of the functional area that the model supports.

Our first activity therefore is to produce an equivalent business model that we like that we can use as the basis for discussion.

Corrective actions include:

1. Create a simple business data model. This should be a model in a Word document that does not include reference data.
2. Produce a short description.
3. Create a glossary of terms.
4. Define the representative business rules.
5. Identify the intended users and the owners of the model.

1.3 Functional Description

In this diagram, arrows point from children to parents. The scope of the data model includes orders for products from customers.

The Functional Description is a simple one-liner:
"Customers issue orders for products that are stored in warehouses".

1.4 A Specific Model in Word

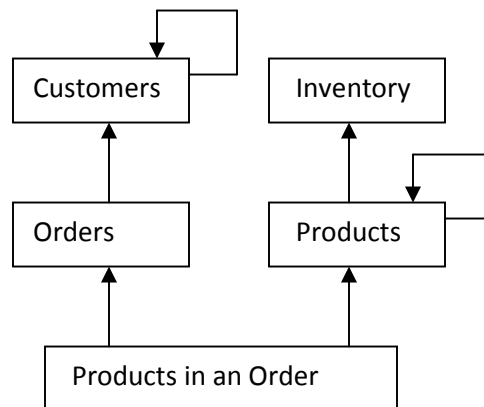The specific version is consistent with the generic version and looks like this:
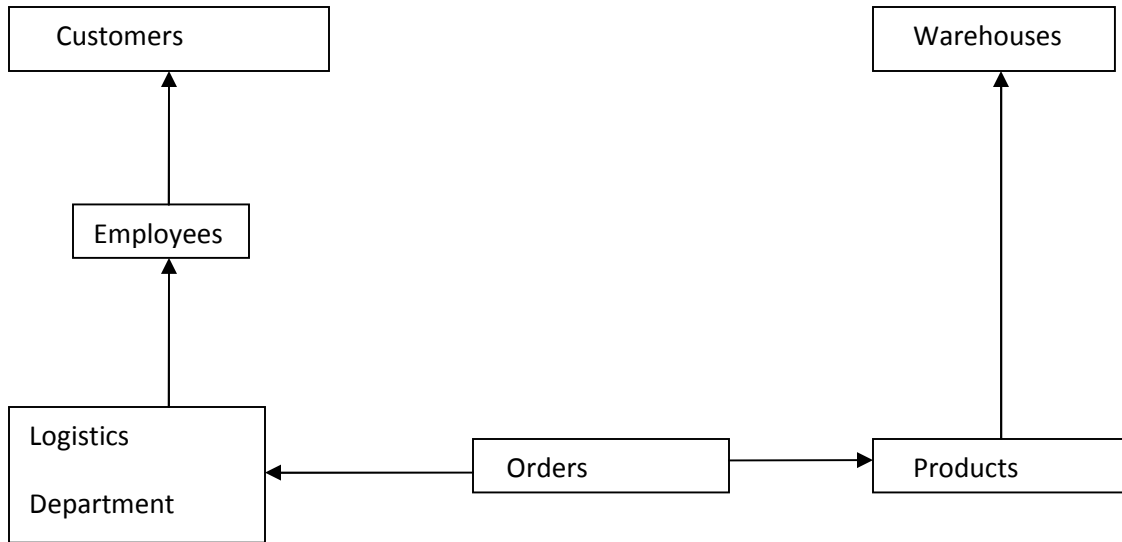
1.5 A Generic Model in Word

In this diagram, arrows point from children to parents.

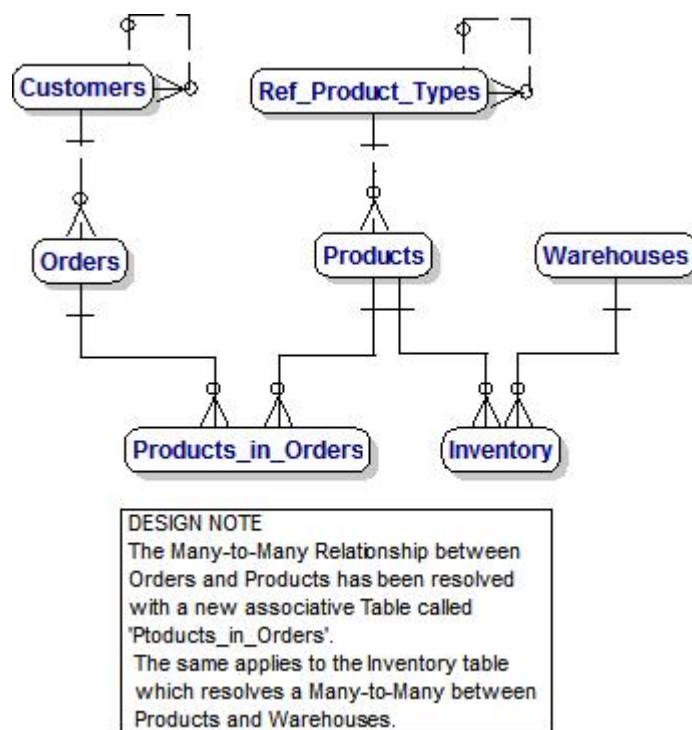Our generic data mode looks like this:

1.6 A Top-Level Generic Data Model

This is a top-level model that was created using a data modeling tool. It shows useful detail, such as details of the relationships. It also replaced a many-to-many with two one-to-many relationships.

This diagram is a generic version of the one below and is useful to help in providing a higher level context for lower-level, more specific models.

An additional level of detail shows the rabbit ears relationship that implements hierarchical relationships for customers and for products.

DESIGN NOTE
The Many-to-Many Relationship between Orders and Products has been resolved with a new associative Table called 'Ptoducts_in_Orders'.
The same applies to the Inventory table which resolves a Many-to-Many between Products and Warehouses.

This model corrects an error in the original model that we were given.

Step 3. Draft the Business Rules

These Rules must be phrased in unambiguous English. Where possible, the English should make it possible to implement a rule in a data model. For example, Rule 1 makes it clear that there is a one-to-many relationship between a ship and an officer.

1. A customer is staffed with many employees.
2. Purchasing departments raise orders.
3. An order must be authorized by an employee.

4. An employee is assigned to one department at any point in time.
5. An employee is assigned to one or many departments during the course of their career.

Templates

Here is a sample template for business rules:

Nr	RELATES TO	OWNER	DEFINITION
BR.D.1	Customers, Orders	Joe Bloggs	A customer can raise zero, one or many orders.
BR.D.2	Customers, Orders	TBD	An order must be associated with a valid customer.
BR.D.3	Orders, Products	Joe Bloggs	An order can refer to one or many products.
BR.D.4	Orders, Products	Joe Bloggs	A product can appear in zero, one or many orders. Therefore, there is a many-to-many relationship between orders and products.

Step 4. Draft a Glossary of Terms

This is a sample template.

TERM	AUTHOR	DEFINITION
Customer	Joe Bloggs	Any customer that can raise an order.

Order	Joe Bloggs	A request for products to be supplied. The format of a request can be a paper document, an online form and so on.
Product	TBD	An item that can be supplied on request. It can be something small, like a pencil, or something large like a printer.

Step 6. Check that the Data Model is Correct

The rules will help in determining whether the model is correct.

In this case, there is an error in that ships are shown coming between ships and departments. The reality is that departments exists without officers. This is corrected in the top-level data model in Section 1.9.

Step 7. Review with Users

Review and revise as necessary.

Step 8. Check Normalized Design

The design looks normalized and therefore is acceptable. The reference data looks appropriate and is not related and therefore is acceptable.

Step 9. Look for Design Patterns

This business model shows these examples of design patterns:

- A one-to-many relationship between ship and office.
- A many-to-many relationship between requisition and product.

It does not show inheritance but in general we would not expect to find it.

There are a number of reasons why inheritance does not appear.

For example:

- Inheritance is not appropriate in this case.
- Inheritance does not show in a data model for a physical database.

Step 10. Review any Data Warehouses

In this case study, this step is not necessary because we do not have a data warehouse or data mart.

Step 13. Check Naming Standards

Standards that are common include:

1. Initial capitals with lower case elsewhere – for example, Customer_id
2. All capitals – for example, CUSTOMER_ID
3. Lower case everywhere – for example customer_id

Any of these standards is acceptable.

If no standard has been established, then number 1 is recommended.

Step 14. Check for Consistent Data types

This check requires a physical data model or some other document that includes this level of detail. The procedure then is to visually scan the documents or use the domain feature of the modeling tool or perhaps SQL to look for discrepancies.

The domain feature allows you to define standard data types for any data item that occurs frequently and then use this domain for every occurrence of the data item.

For example, a name could be defined as a variable-length character string with a length of 258.

Then whenever a name appears in a model, the modeler can select this domain as convenient shorthand and also a simple way to enforce consistency.

It will be necessary to analyze any discrepancies and decide on a standard to resolve them.

Step 11. Check for Defaults

Default values are a powerful technique for adding values in a data model. They can be used to enforce consistency.

Probably the most common example is to specify that the date of entry and creation of a new record should be the current system date.

This applies to new customers, orders and the date of any payment or adjustment and so on.

Step 15. Determine the Assurance Level

Appendix A defines the process to be followed and discussed appropriate remedial follow-up action.

1.7 What have we learned?

We have learned some very important rules that will help us to assess the quality of a data model created by someone else.

They are a combination of theory based on Dr. Codd's original work at IBM (see http://www.databaseanswers.org/codds_page.htm) and some best practice guidelines based on the experience that the Database Answers team has gained over the last few decades.

(This is the back cover)

Barry Williams is the founder and principal consultant with Database Answers.

His company has been providing advice and assistance to a wide range of blue-chip clients for over 20 years.

His particular interest is in advancing the role of data models as a way of improving communication between the business user community and data management professionals.

As part of this role he publishes best practice on his Database Answers Web site at

- http://www.databaseanswers.org/