

Table of Contents

Welcome	4
Volume Three : Advanced Level	5
13. From the Cradle to the Grave	5
14. Check the Quality of a Data Model	30
15. Enterprise Data Models.....	47
16. Master Data Management (MDM)	85
17. Build an Enterprise Data Model.....	107
18. A Single View of the Truth	133
19. A Case Study	144

Data Modeling by Example

Volume Three

Barry Williams

Database Answers Ltd.

comments@databaseanswers.org

First Kindle Original Edition: London, 2012

Thank you to these great people for their valuable comments on early drafts of this book.

USA:

Cary Stiebel, US Army, Fort Ord, Monterey Bay, California

Mauricio Caneda, New York, NY

Matt Baugh, Idaho Falls, Idaho

Sandra Baia Overton, New York, NY

Slawomir Pazkowski, Eagan, Minnesota

Other parts of the world:

Andy Cheng, Chengdu, China

Arundhati Pawaskar, India

Benjamin Mortensen, Copenhagen, Denmark

Daniel Petterssen, Stockholm, Sweden

Erik Wahlfelt, Copenhagen, Denmark

Glen Michael, Kirriemuir, Scotland

Julian Apatu, New Zealand

Manus le Roux, Damelin Vaal, South Africa

Murat Kalin, Istanbul, Turkey

Nick Walsh, London, England

Seshi Reddy Bejawada, India

Soren Andresen, Roskilde, Denmark

Welcome

We have produced this book 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 the Web site:

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

Why is This Book Free?

This book is free for a limited period to encourage feedback and suggestions for things the author can do to improve the book.

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

Barry Williams

Principal Consultant

Database Answers Ltd

London

England

Volume Three : Advanced Level

Finally, we come to the challenging topics of enterprise data models and master data management (MDM), how to combine existing data models to create a new one and using models to establish a single view of the truth.

13. From the Cradle to the Grave

13.1 Introduction

This chapter will discuss data models that are appropriate to the stages in our lives from the earliest to the latest.

13.1.1 What is This?

It is structured as a tutorial that takes you step-by-step through each stage and discusses how you create data models at each stage.

13.1.2 Why is it Important?

This is important because it helps you understand, starting from first principles, how to deal with increased complexity and conform to the basic principles of a well-designed data model.

13.1.3 What Will I Learn?

The approach in this chapter is to discuss data models covering a typical life cycle from a new-born baby to an old person. This allows us to trace the increasing complexity in life and match it to an increasing complexity in data models.

The approach has three steps:

1. Establish the scope of the data model
2. Identify the 'Things of Interest' that are within the scope. These will be called *entities*.
3. Determine the relationships between them.

Establishing the Scope of our Data Model

We have decided that the scope is the 'passages' in our lives from the cradle to the grave.

This will include childhood, teenage years, becoming a student, getting a job, getting married, getting sick, and finally dying

Therefore, any items outside this scope are not '**Things of Interest**'.

Topics covered:

- Primary Keys and Foreign Keys
- One-to-Many and Many-to-Many Relationships
- Hierarchies and Inheritance
- Reference Data

13.2 I am a New Baby

Topics covered include:

- Entities
- Primary Keys



Baby Elephant in Africa



During the early days, the baby is aware only of its own existence.

In Sweden, new-born babies are issued with unique National Identity numbers.

13.3 Me and Mommy

At this stage, the baby becomes aware of Mommy's existence so we add her to the model because she is now in scope.

Topics covered include:

- Foreign Keys



Baby Elephant and Mommy

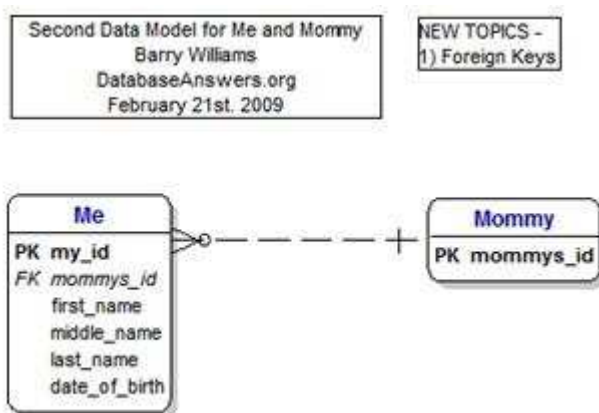
Here we add **relationships** between the entities. When this primary key is used in another table, it is referred to as a *foreign key*. We can see a good example in this diagram, where the 'mommys_id' field appears in the Me Table as a foreign key. This is shown with an 'FK' symbol beside it. The 'mommys_id' field then appears as the primary key in the Mommy Table.

Mandatory Key Fields

A foreign key is usually **mandatory**, in other words, a value for a mommys_id in the Me Table must correspond to the value of the mommys_id for a record in the Mommy Table.

In plain English the business rule would say "Each baby must have a real mommy."

This is shown in the diagram by the short straight line at the end of the dotted line close to the Customers Table.



13.4 Me, Mommy, and Meals

Now I become aware that I am eating at regular times.

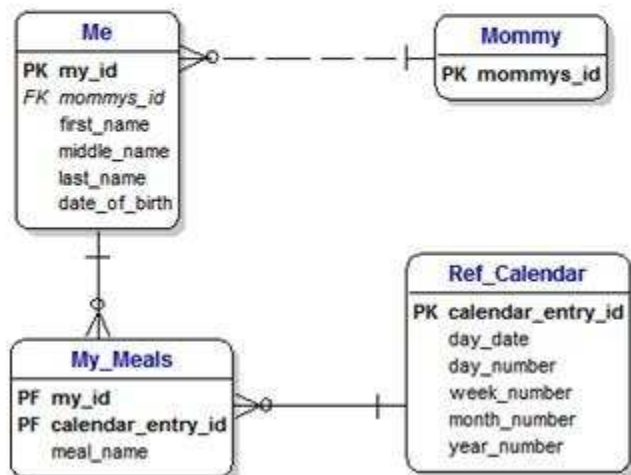
Topics covered include:

- One-to-Many Relationships



Third Data Model for Me and Mommy
Barry Williams
DatabaseAnswers.org
February 21st, 2009

NEW TOPICS -
1) Reference Data
2) Many-to-Many Relationships



13.5 Children's Playgroups

Topics covered include:

- Many-to-Many Relationships

Here we have added the **relationships between** the entities.

When this primary key is used in another table, it is referred to as a *foreign key*.

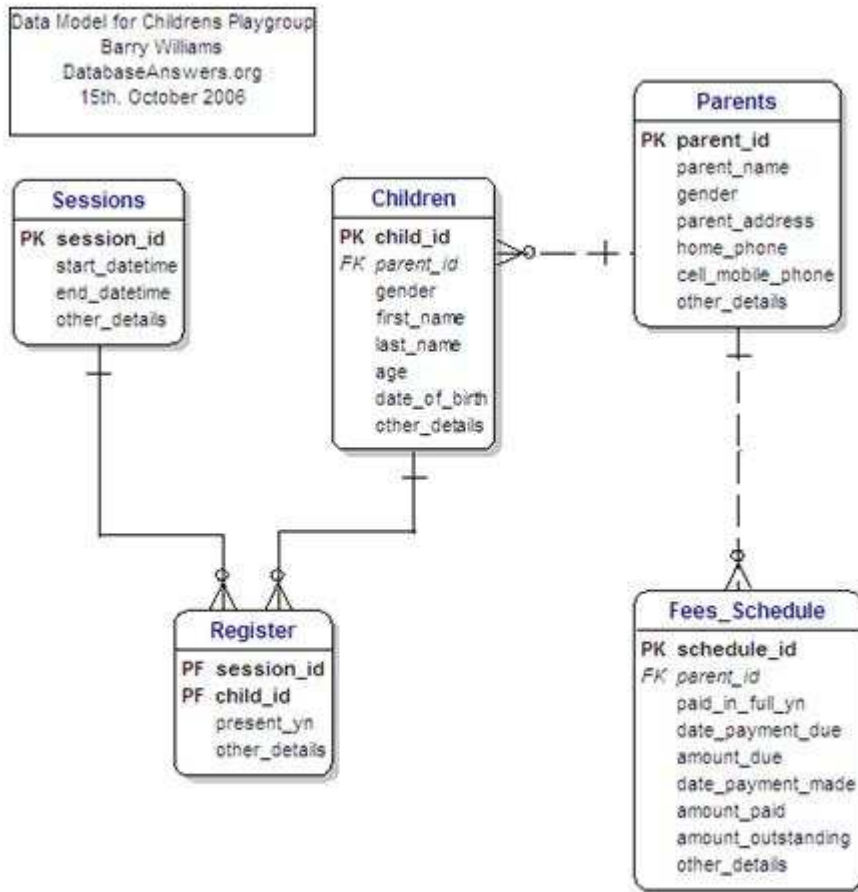
We can see a good example in this diagram, where the Customer_ID appears in the Customers_Payment_Methods Table as a foreign key.

This is shown with an 'FK' symbol beside it

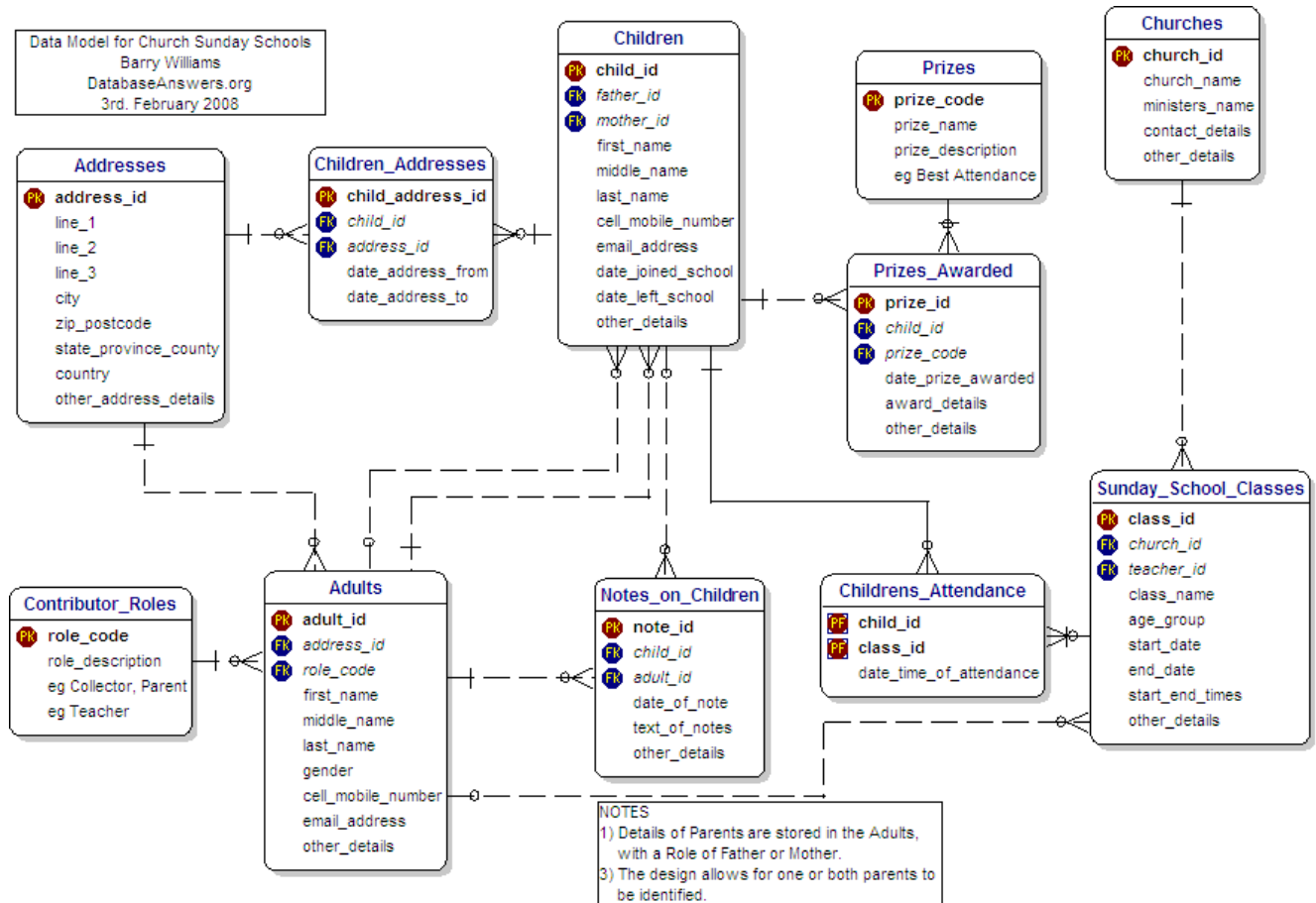
Mandatory Key Fields

A foreign key is usually mandatory, in other words, a value for a Customer_ID in the Customers_Payment_Methods Table must correspond to an actual value of the Customer_ID in the Customers_Version_1 Table.

This is shown in the diagram by the short straight line at the end of the dotted line close to the Customers Table.



13.6 Church Sunday School



13.7 Student Accommodation

At this stage, I move into student accommodation.

Topics covered include:

- Primary and Foreign Keys
- One-to-Many and Many-to-Many Relationships
- Reference Data

This diagram shows how the hierarchies of products and product types that we have just discussed are shown in our **Entity-Relationship** diagram.

Rabbit Ears

You will notice that the table called 'Product_Types_v1' has a dotted line coming out on the right-hand side and going back in again on the top-right corner.

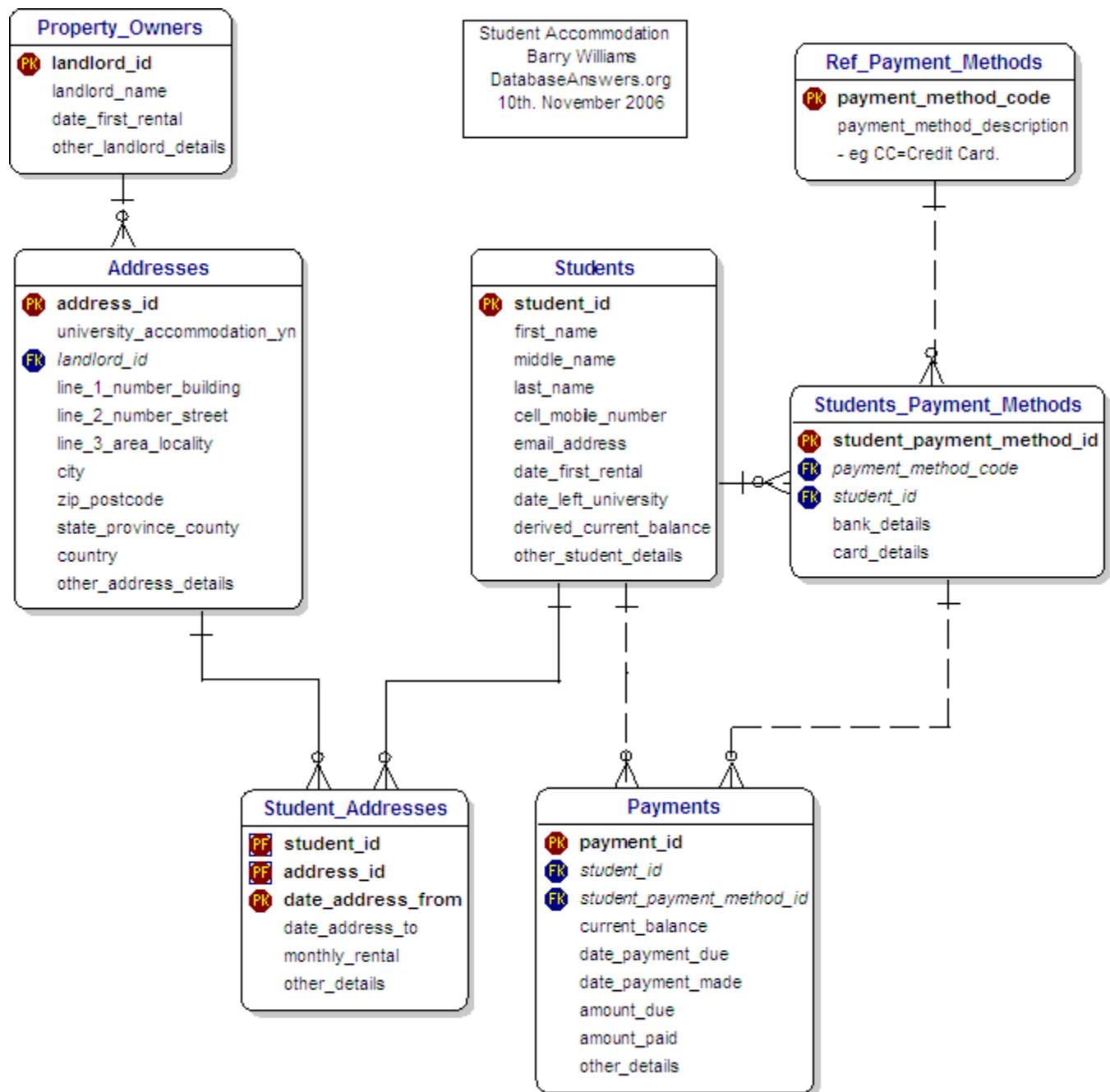
Data analysts call this a *reflexive* relationship, or informally, simply *rabbit ears*.

In plain English, we would say that the table is joined to itself and it means that a record in this table can be related to another record in the table. This approach is how we handle the situation where each product can be in a hierarchy and related to another product.

For example, a product called 'Panini' could be in a product sub-category called 'Miscellaneous Sandwiches,' which could be a higher product category called 'Cold Food,' which itself could be in a higher product super-category called simply 'Food'.

Next time you go into a coffee shop, take a look at the board behind the counter and try to decide how you would design the products area of the data model.

You should pay special attention to the little 'zeros' at each end of the dotted line. These are how we implement the fact that the 'Parent Product Type Code' is optional, because the highest level will not have a parent.



13.8 Student Assessments

Topics covered include:

- Primary and Foreign Keys
- One-to-Many and Many-to-Many Relationships

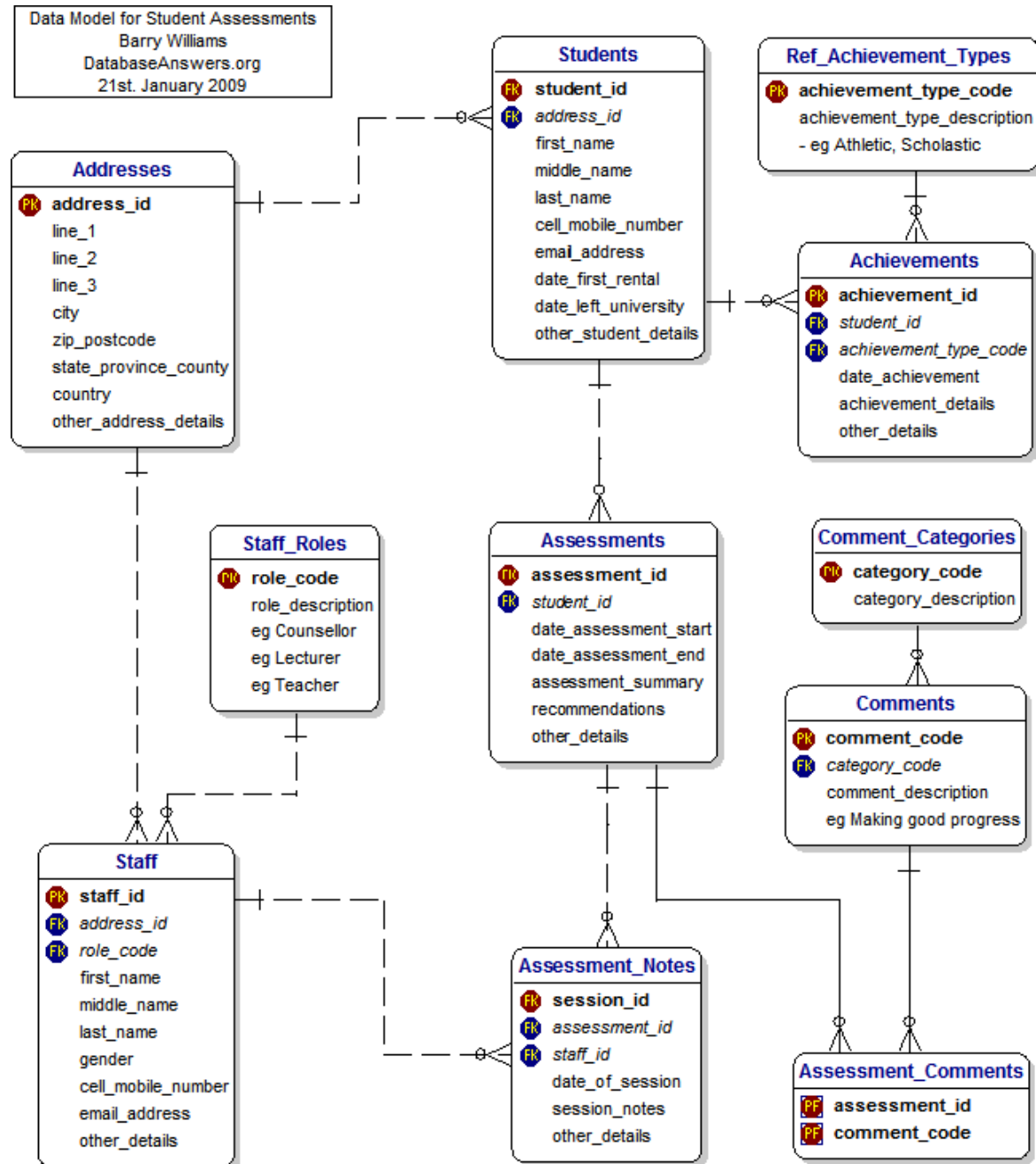
- Reference Data - Addresses

This model shows that:

A student can have zero, one or many achievements

A student can have zero, one or many assessments

Each assessment can be associated with notes.



13.9 Joining Facebook

Topics covered include:

- Primary and Foreign Keys
- One-to-Many and Many-to-Many Relationships
- Reference Data

This diagram shows address types, which are an example of reference data. This kind of data has the following characteristics:

It doesn't change very much.

It has a relatively small number of values, usually less than a few dozen and never more than a few hundred.

Therefore we can show it with a code as a primary key.

Data in Reference Data Tables can be used to populate drop-down lists for users to select from.

In this way, it is used to ensure that all new data is valid.

13.9.1 Standards

In the Address Table, you will see a field called 'iso_country_codes'.

ISO stands for the 'International Standards Organization'.

Where possible, it's always good to use national or international standards.

13.9.2 Customer Addresses

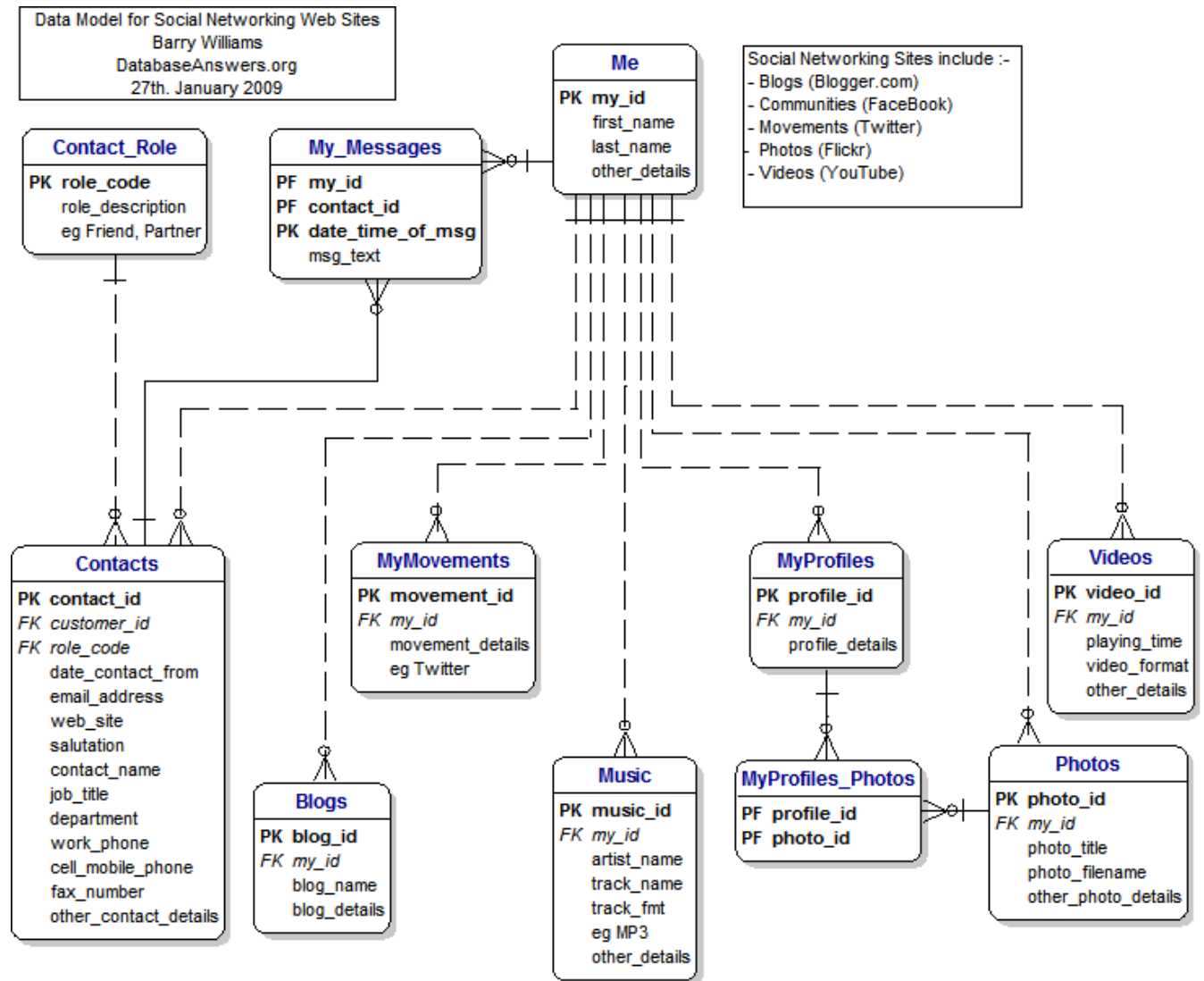
This is a general and flexible approach to handling addresses in our data model. We have a separate Address Table, so we can have more than one address for any customer very easily.

This design also has some other benefits:

We can accommodate more than one person at the same address. We need to do this because different members of a family may sign-up separately with Amazon.

With a separate table of addresses, we can easily use commercial software to validate our addresses. To find this kind of software, simply Google 'Address Validation Software'. The author has used QAS with great success in the past.

With this approach, we can always be sure that we have 100% good address data in our database.



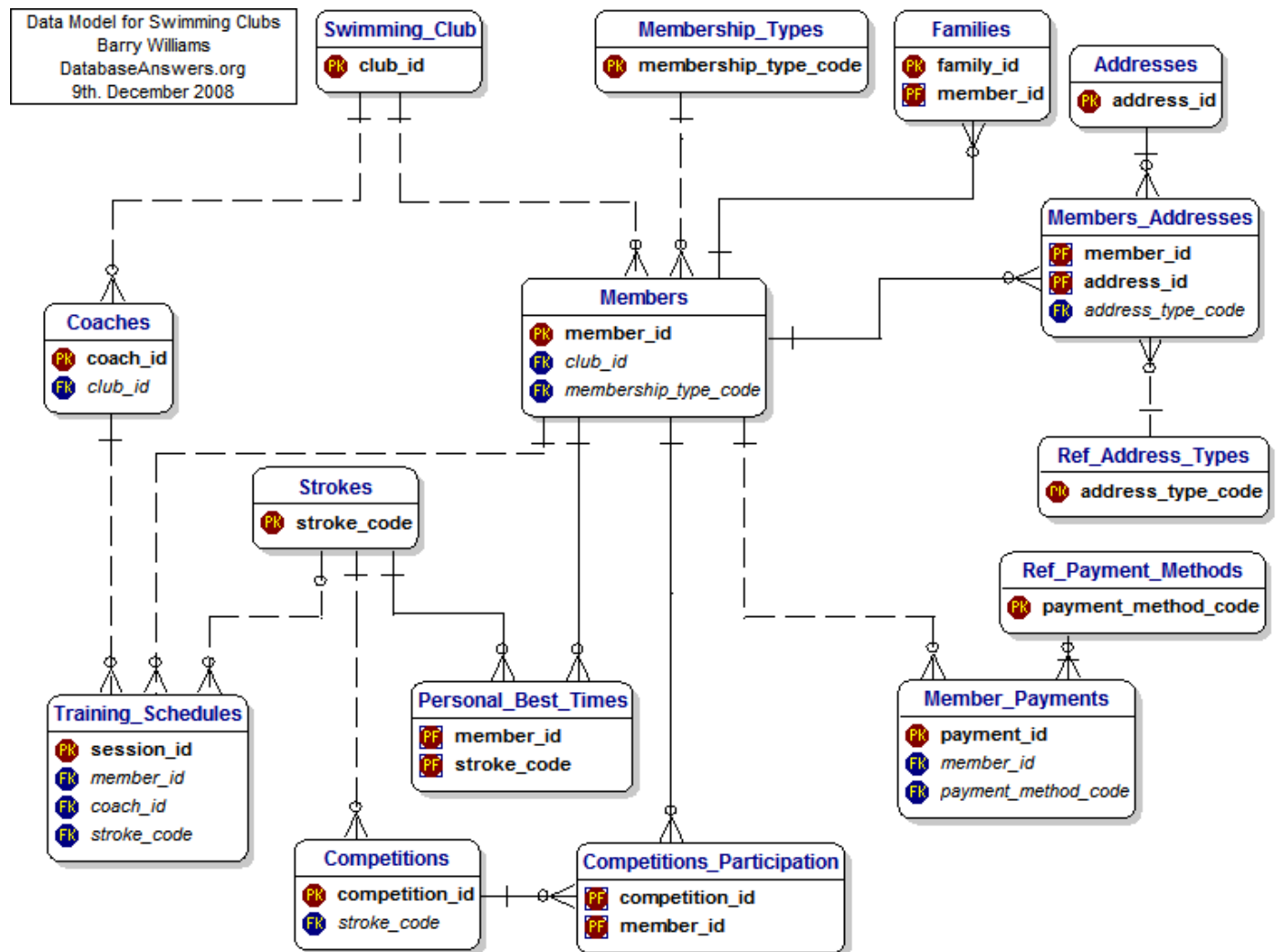
13.10 Joining a Swimming Club



Mission Viejo Masters Competition

2009 U.S. Masters Swimming Club of the Year

The central entity/table in this data model is members, which emphasizes its importance.

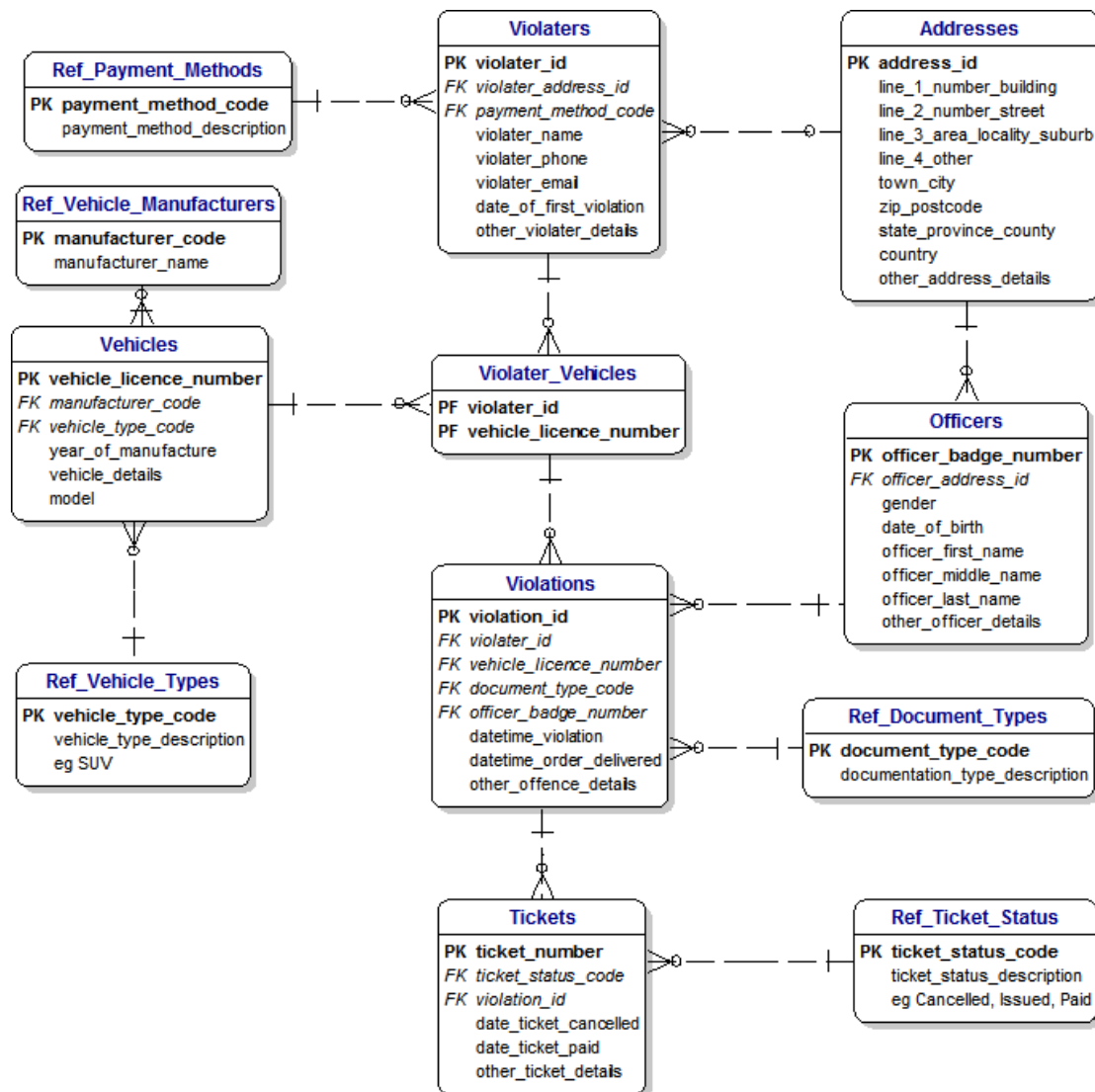


13.11 A Ticket from a Traffic Cop

We start with violators who are always associated with a vehicle. Vehicles in turn are always associated with one or more violations, which result in violations.



Robert Blake as a traffic cop in *Electra Glide in Blue*.



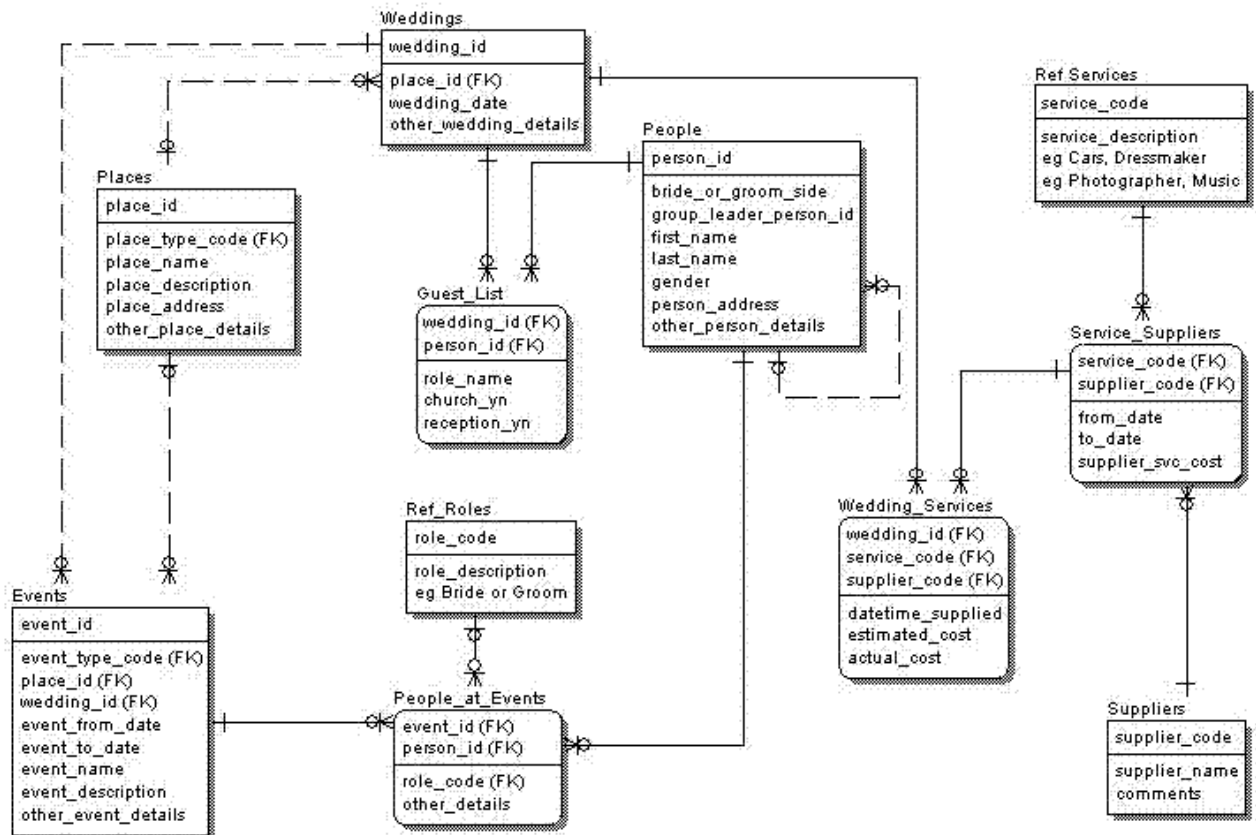
13.12 I Get Married

Topics covered:

- Primary and Foreign Keys
- One-to-Many and Many-to-Many Relationships
- Reference Data

This model was created using a different data modeling tool, called ERWin from Computer Associates. It shows that if you are familiar with the underlying principles that you will be able to understand and ERD.

The **wedding** is the dominant table and is held in one place.



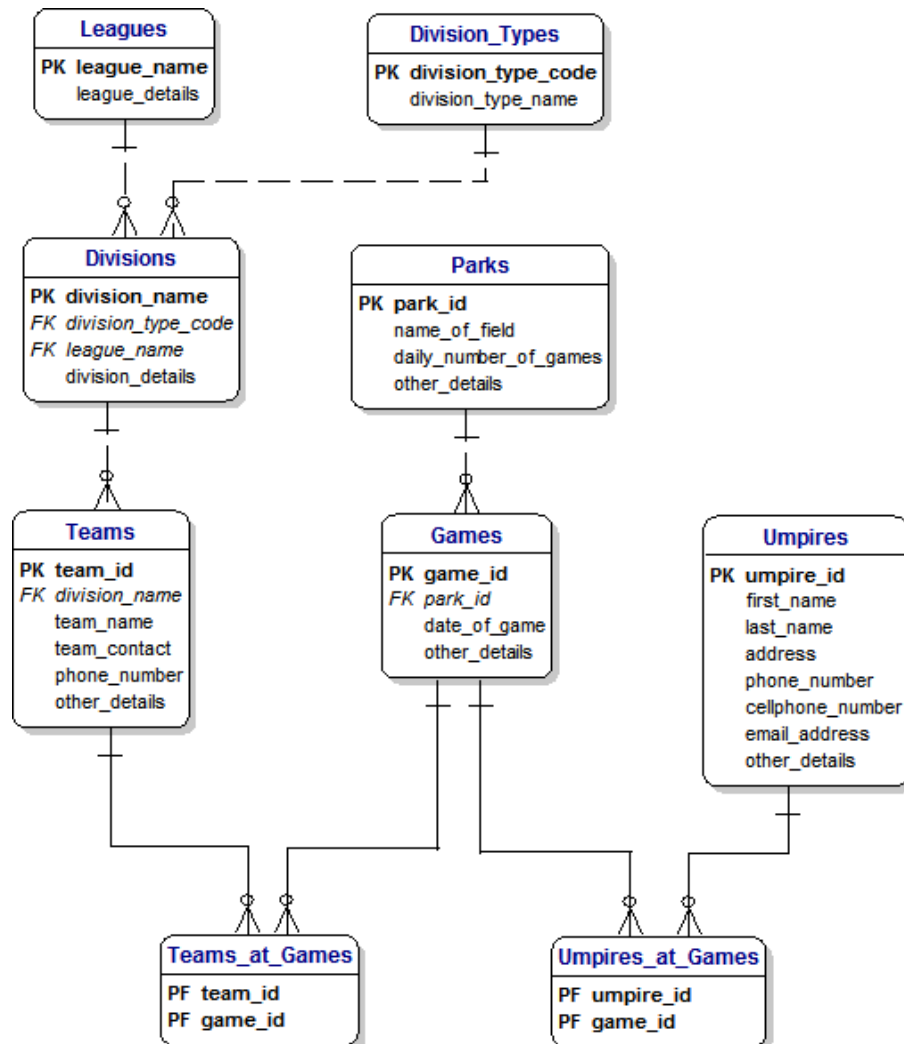
13.13 I Become a Baseball Umpire



Baseball Umpire Calling a Strike

Topics covered:

- Primary and Foreign Keys
- One-to-Many and Many-to-Many Relationships
- Reference Data



13.14 I Go to Hospital



Massachusetts General Hospital, Boston, Mass. USA

Here we see that **patients** is the dominant table.

The rules are:

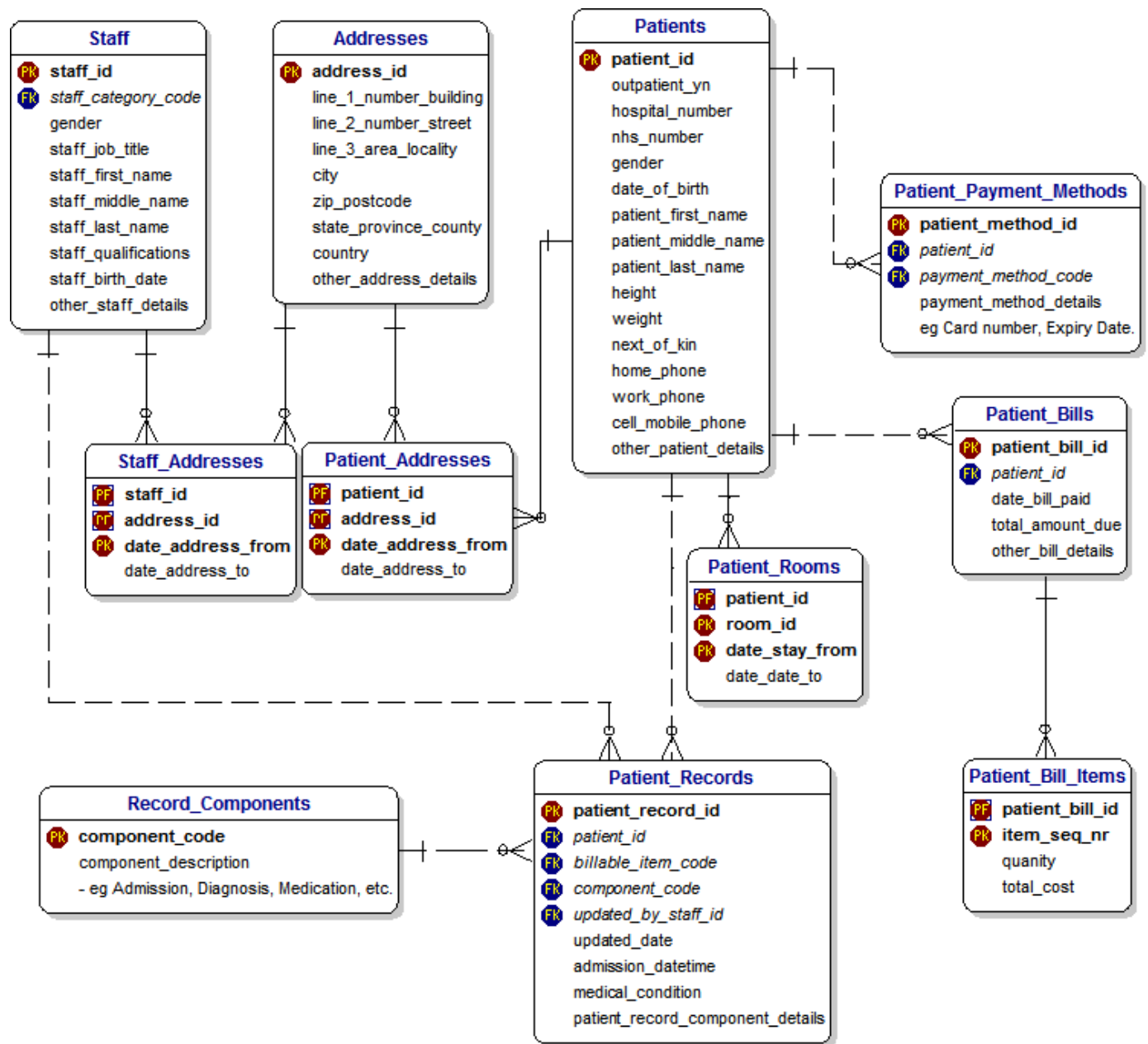
Each patient can have many addresses.

Each patient can have zero, one or many payment methods.

Each patient can have zero, one or many patient bills.

Each patient can be allocated to zero, one or many rooms.

In the US, bills are an important part of a trip to the hospital. This is not the case in Europe and other parts of the world.



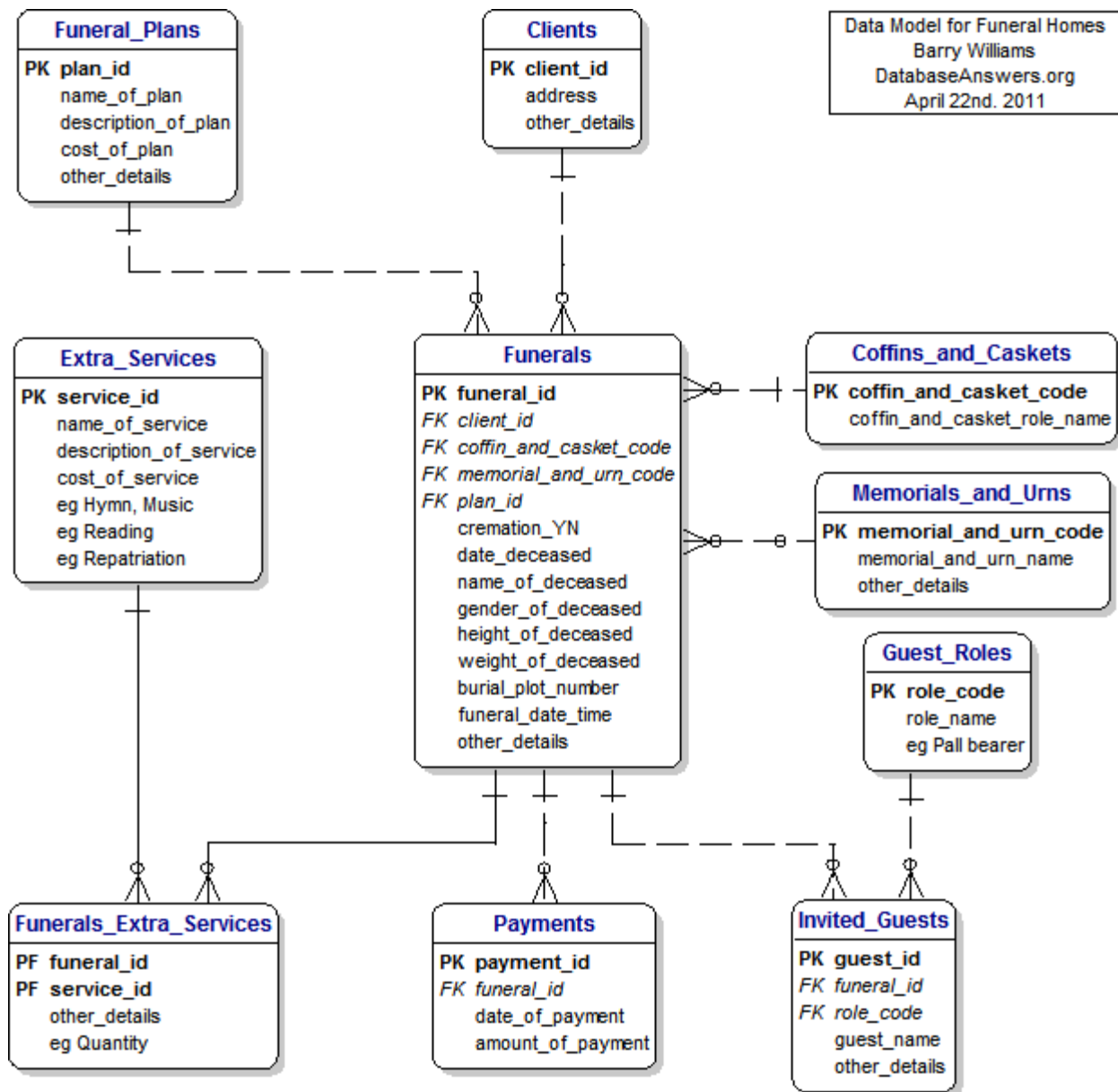
13.15 I Visit a Funeral Home



Kelly's Funeral Home, Ottawa, Ontario, Canada

The author lived in Ottawa for six great years.

Here we can see that the dominant thing is **funerals**. Every funeral must be associated with a client and has a funeral plan.



13.16 Events in my Life

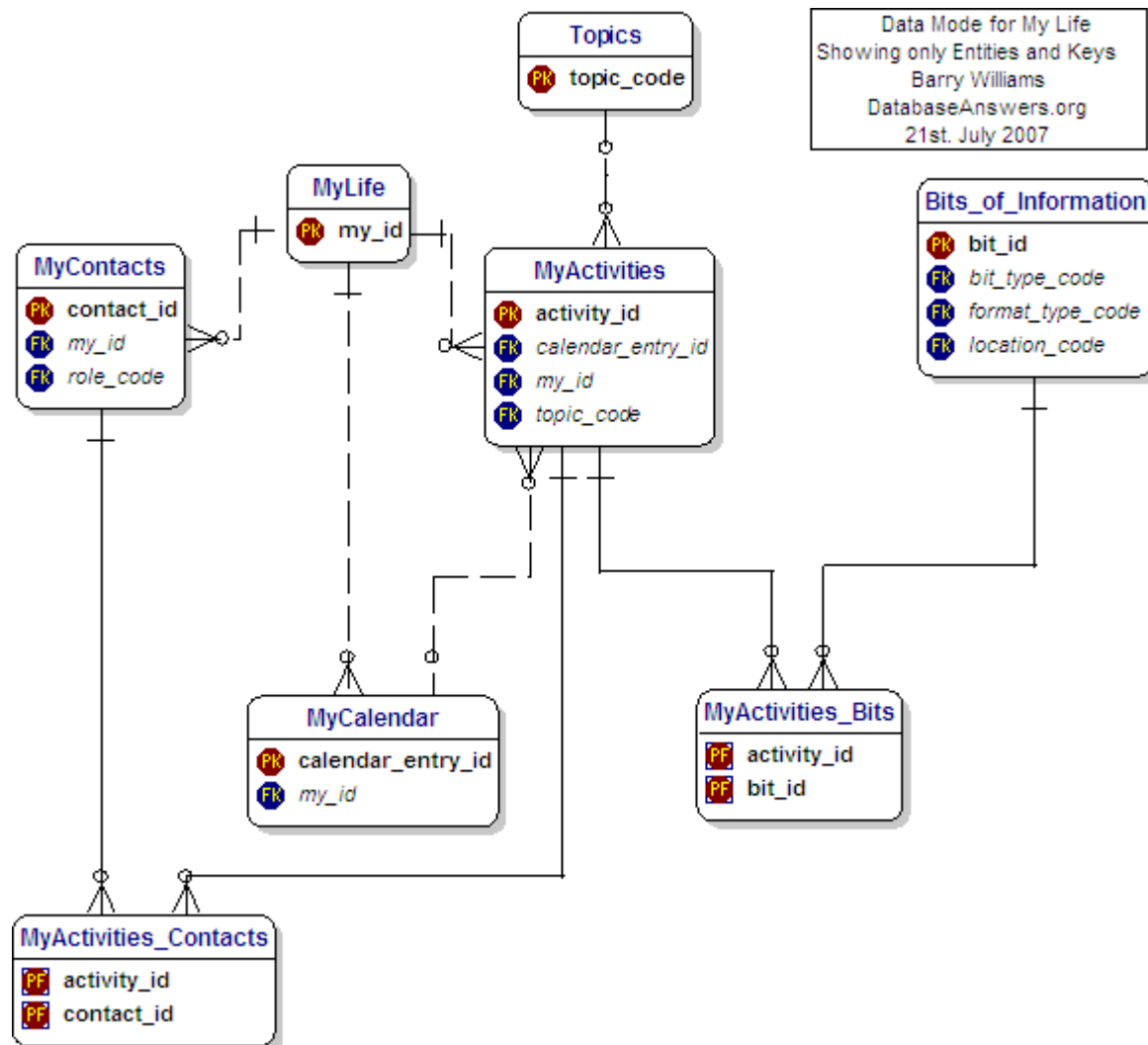
Topics covered:

- Primary Keys
- Foreign Keys
- One-to-Many Relationships
- Many-to-Many Relationships

- Reference Data

This is based on the 'My Life' data model:

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



13.17 Events in my Work

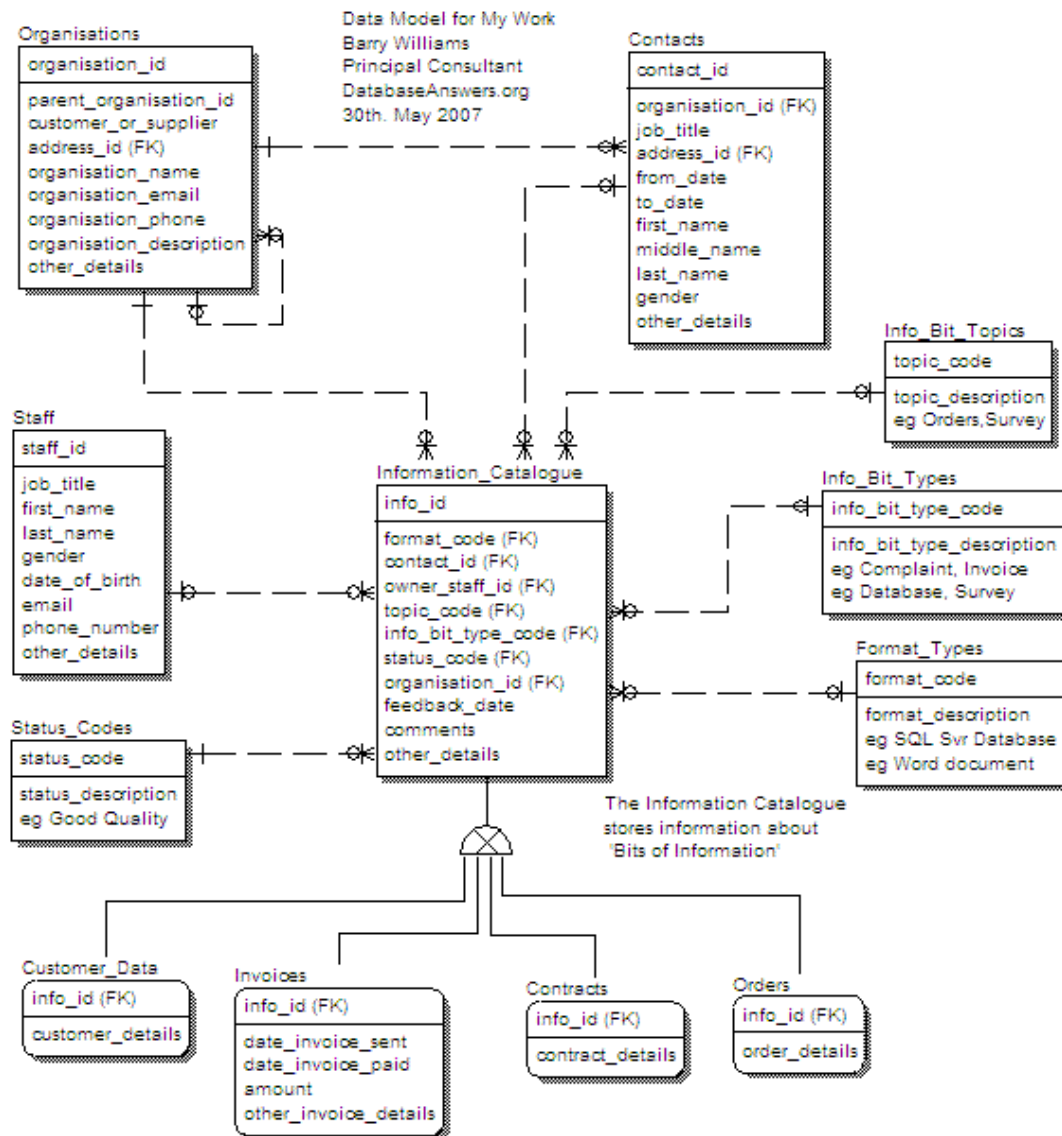
Topics covered:

- Primary Keys and Foreign Keys
- One-to-Many and Many-to-Many Relationships

- Hierarchies (e.g. Organizations) and Inheritance
- Reference Data (e.g. Status Codes)

This is based on the 'My Work' data model:

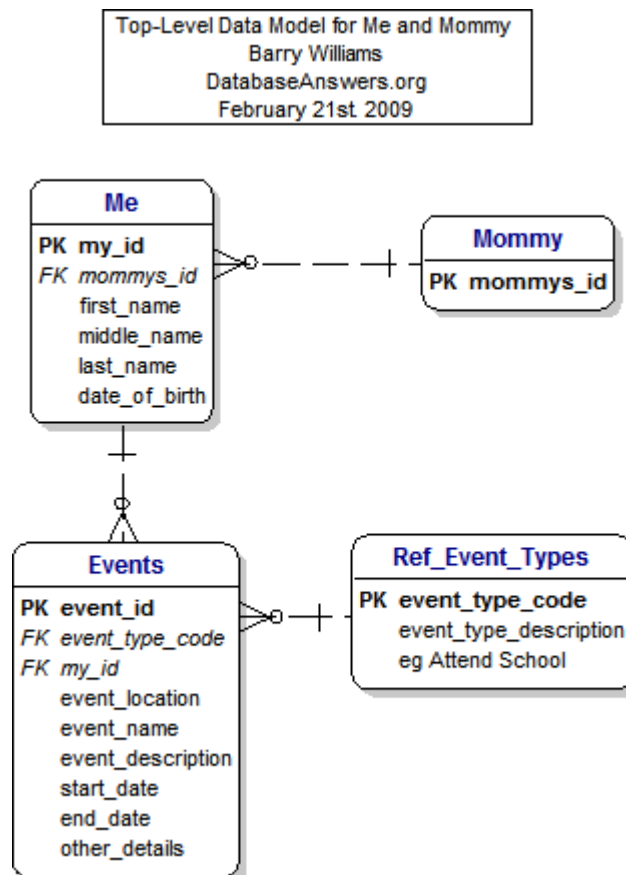
- http://www.databaseanswers.org/data_models/my_work/index.htm



13.18 Canonical Data Model

This is a beautifully simple model which shows that Mommy is the single constant factor and that 'My Life' is a series of events of different types.

A **canonical** data model is one that is stripped of everything superfluous.



13.19 What Have We Learned?

This chapter aims to bring together a number of data models that cover things that we are all familiar with. Our purpose in bringing them together is to present a range of topics that become increasingly complex in a way that should help us to understand this complexity.

14. Check the Quality of a Data Model

14.1 Introduction

14.1.1 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.

14.1.2 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.

14.1.3 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.

14.2 Create a Top-Level Business Data Model

14.2.1 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.

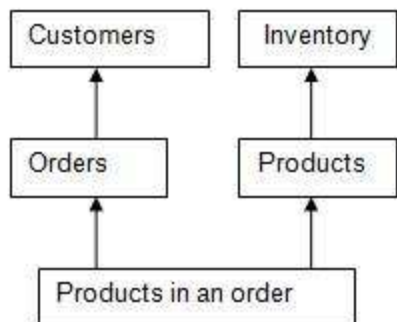
Logical models will include primary and foreign keys and often the modeling tool will provide a facility to generate a physical model from a logical one.

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

14.2.2 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.

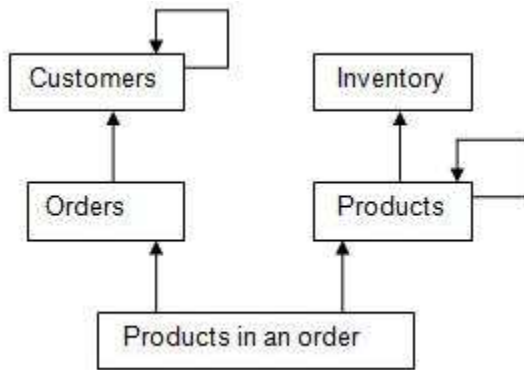
This diagram is a good example:



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.



14.3 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 'commCustomer' will then be able to agree and sign off the rules.

Here is a small example:

Nr	TABLE	DESCRIPTION
D.1	Orders and Customers	<p>An order must be raised by a valid customer.</p> <p>Not every customer will raise an order.</p> <p>Therefore the relationship must be one-to-many with a mandatory condition at the customer end and an optional condition at the order end.</p>
D.2	Orders and Products	<p>An order must refer to valid products.</p> <p>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>
P.1	Products and Inventory	<p>Products are kept in store's inventory.</p>

14.4 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.	

14.5 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.

14.6 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.

14.7 Check Normalized Design

14.7.1 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?

14.8 Reference Data

14.8.1 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.

14.8.2 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.

14.9 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.

14.10 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?

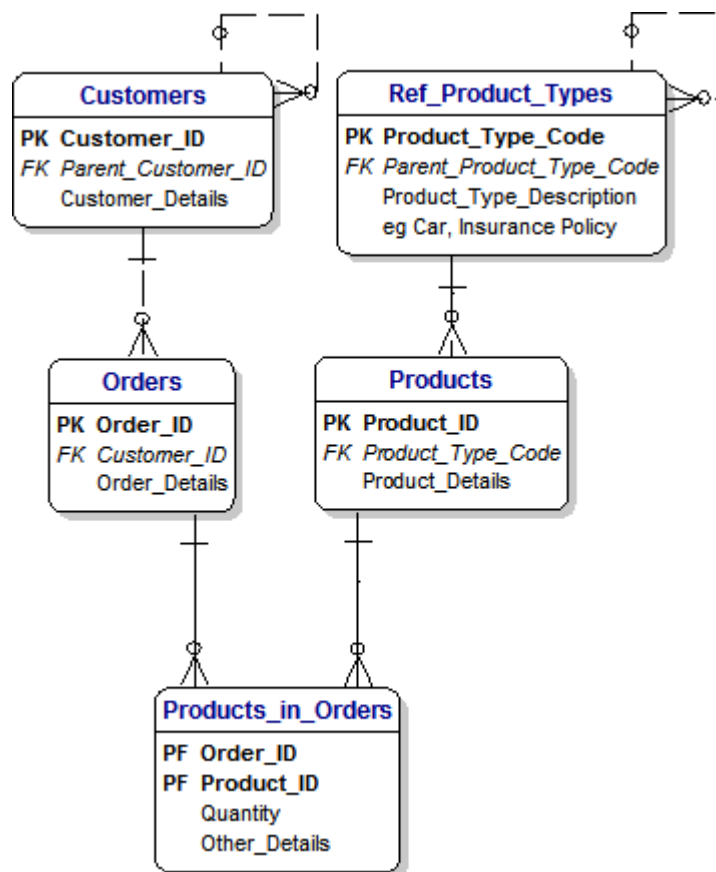
14.11 Look for Design Patterns

14.11.1 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.



14.11.2 Inheritance

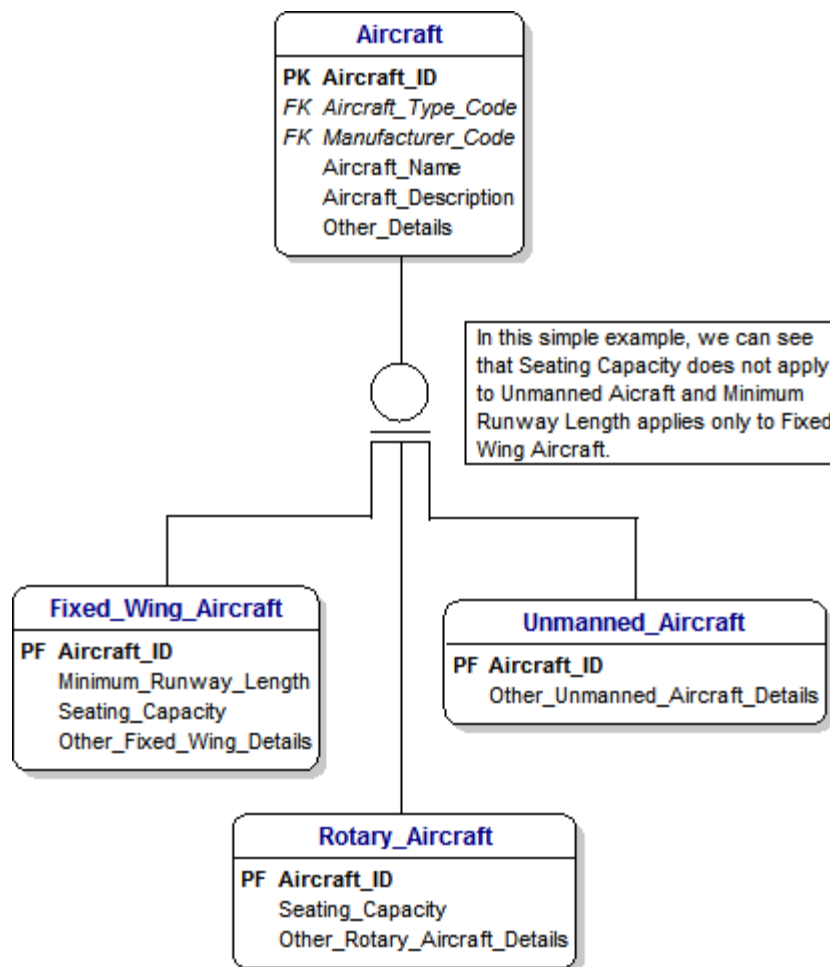
More details are provided in Chapter 3. Concepts in the document entitled "How to Understand a Data Model".

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:



14.11.3 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.

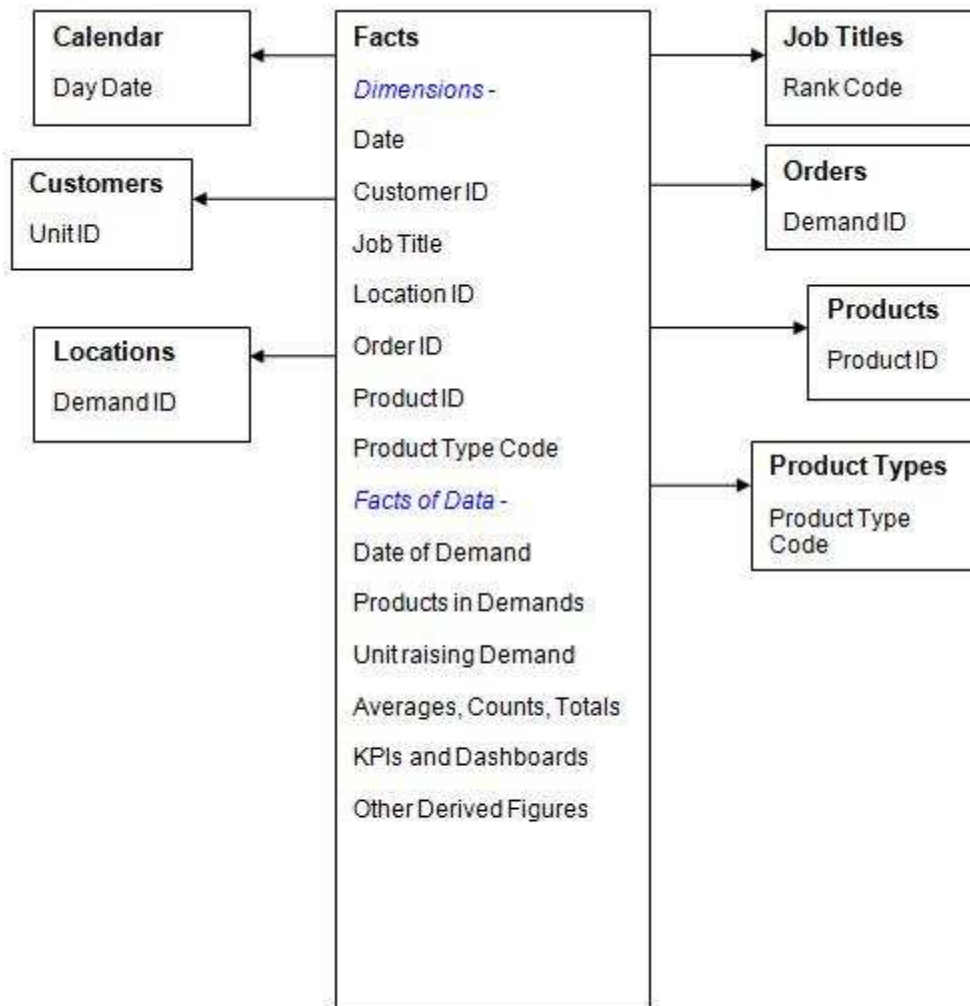
For example, when a soldier joins the army there might be data that is involved only with the joining details. The basic data for the soldier will be part of his or her basic records – such as date of birth and place of birth. If a separate table exists for 'Joining Details' then it would contain such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

14.12 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.



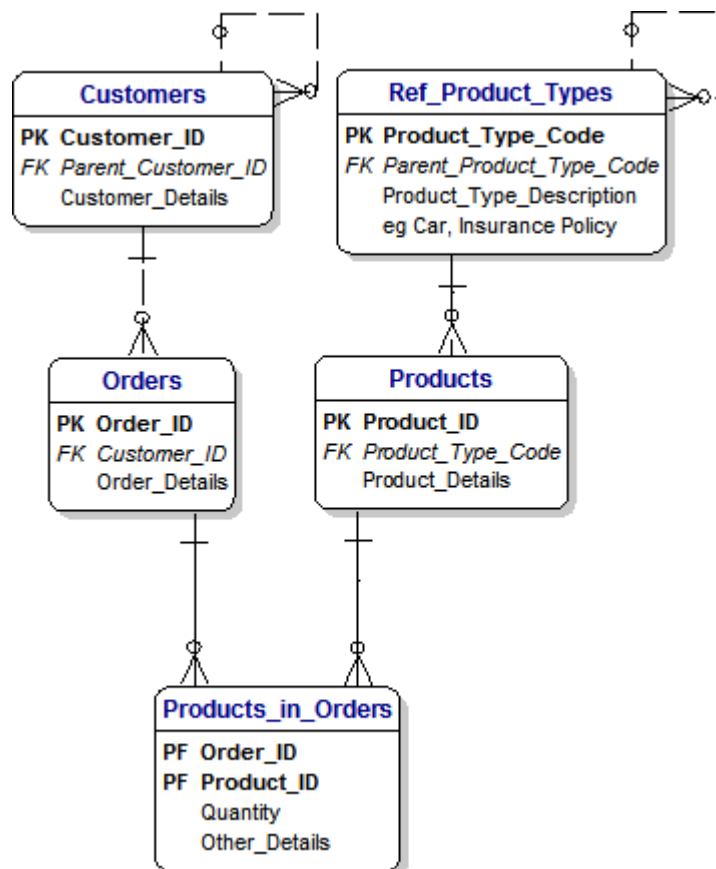
14.13 Check Naming Standards

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.



14.14 Check for Consistent Data Types

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

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

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*.

14.15 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'.

14.16 Determine the Assurance Level

The assurance level could be:

Acceptable

Acceptable with reservations

Not acceptable

14.17. Checklist for Quality Assurance

This Checklist extends the basic concept of the data model scorecard that was originated by Steve Hoberman.

FEATURE	VALUE	COMMENT
Can the scope of the model be defined?	Essential	There must be clear alignment of the model with the business and the user requirements.
Can the requirements be defined?	Essential	Requirements must be defined.
Does the model meet the requirements?	Essential	The model must meet the requirements.
Is there a comprehensive glossary of terms?	Essential	Very important that the value of a glossary is recognized.
Have comprehensive business rules been defined?	Essential	Very important that the value of rules is recognized.
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) 5. Does the same item of data appear in more than one table? (N)	Desirable	The model might be OK even if it fails all of these checks. The power and flexibility of the relational approach makes it possible to handle all sorts of errors and still provide the foundation for an operational database.
Is there compliance with any relevant data standards?	Desirable	Not essential.
Does the model use relevant design patterns?	Desirable	Might be OK even if design patterns can't be identified. But if it isn't laid out well it brings into question the professionalism of the creators of the model.
Is the model easy to read and understand?	Desirable	The flow of logic in a data model should go from top-left to bottom-right. It might be OK even if it can't be read and understood. But if it isn't laid out well it brings into question the professionalism of the creators of the model.
Can any repeating groups be identified?	Not critical	Not critical because the database will work OK. Usually indicates denormalization for performance reasons. In other words, this can be acceptable for a database design. For example, address lines often repeat.
Can any derived data be identified?	Not critical	Not critical because the database will work OK.
Does the model follow naming standards?	Not critical	Cosmetic but is a measure of the professionalism.

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.

14.17.1 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.”

14.17.2 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.

14.17.3 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.

14.17.4 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.

14.17.5 For Data Migration:

It is essential that the model is correct at the detailed level of tables, fields and data types.

14.18 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.

15. Enterprise Data Models

15.1 Introduction

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

15.1.1 What is This?

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

15.1.2 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.

15.1.3 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.

15.2 Business Intelligence (BI)

15.2.1 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:

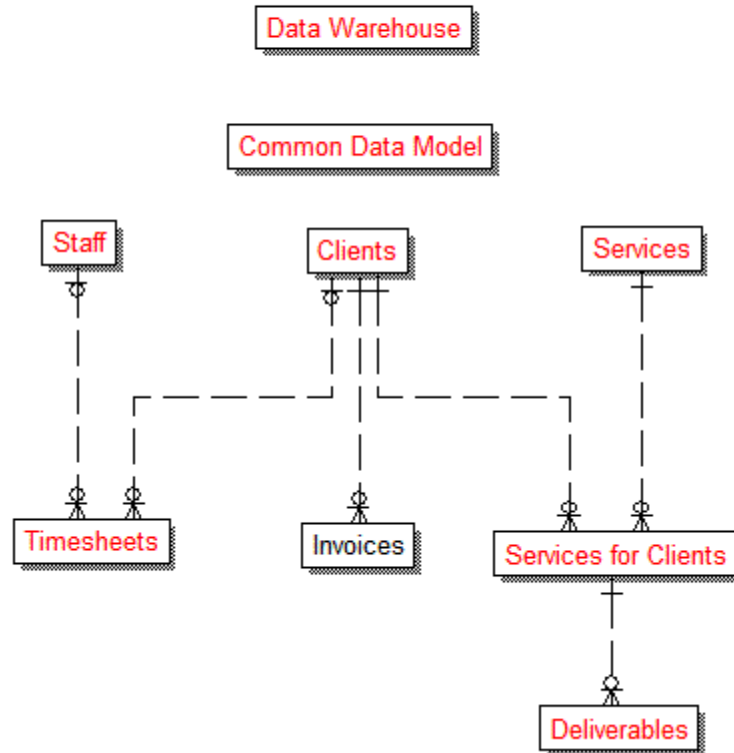
- Data warehouse and data marts
- Key Performance Indicators (KPIs)
- Data sources and integration

15.2.2 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

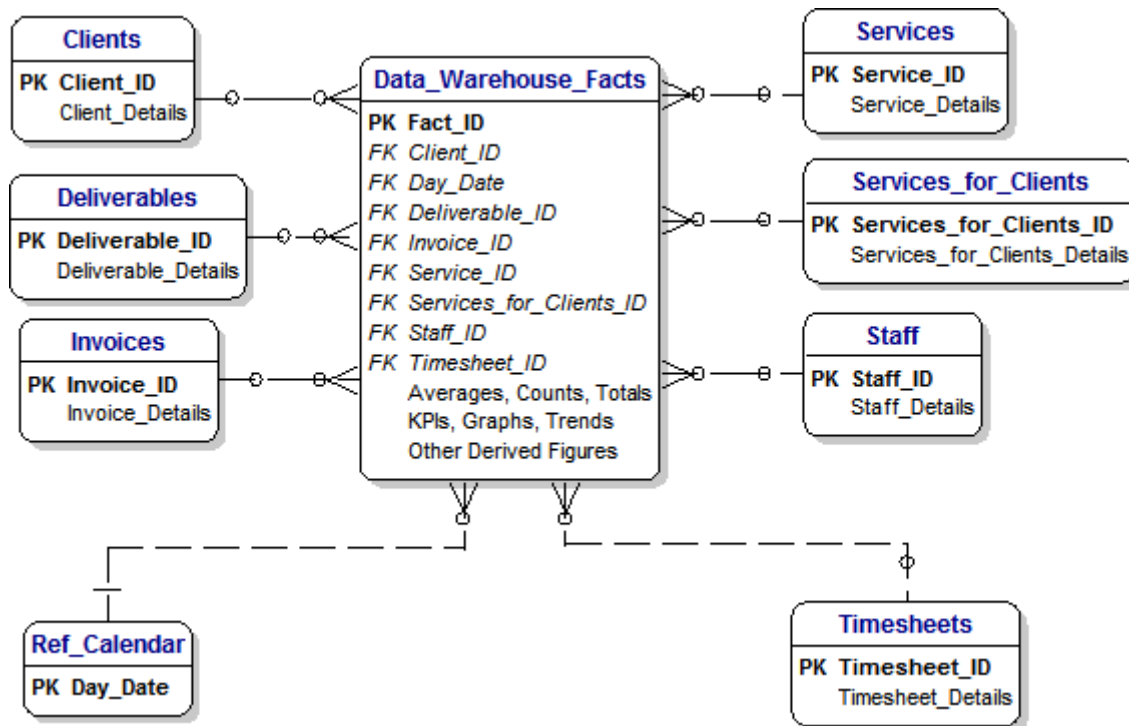
Enterprise Data Model for BMEWS BI
Barry Williams
Principal Consultant
DatabaseAnswers.com
April 19th 2011



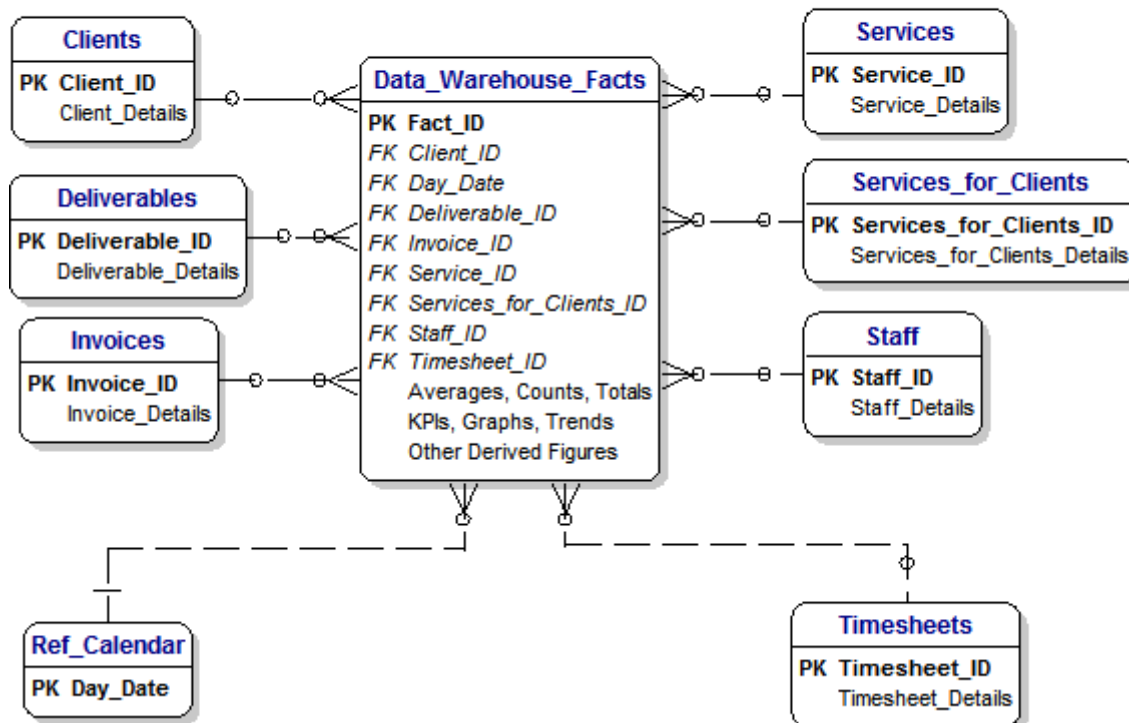
15.2.3 BI Data Warehouse

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



15.3 Insurance

15.3.1 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.”

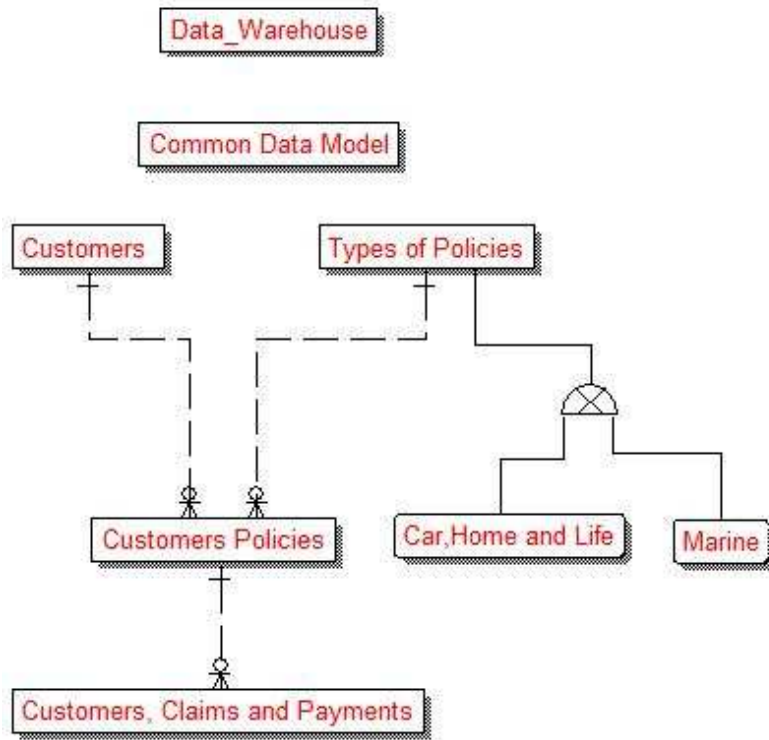
Therefore, the subject areas in the enterprise data model must include:

- Claims
- Customers
- Policies
- Products and services
- Staff
- Settlements

15.3.2 The Enterprise Data Model

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

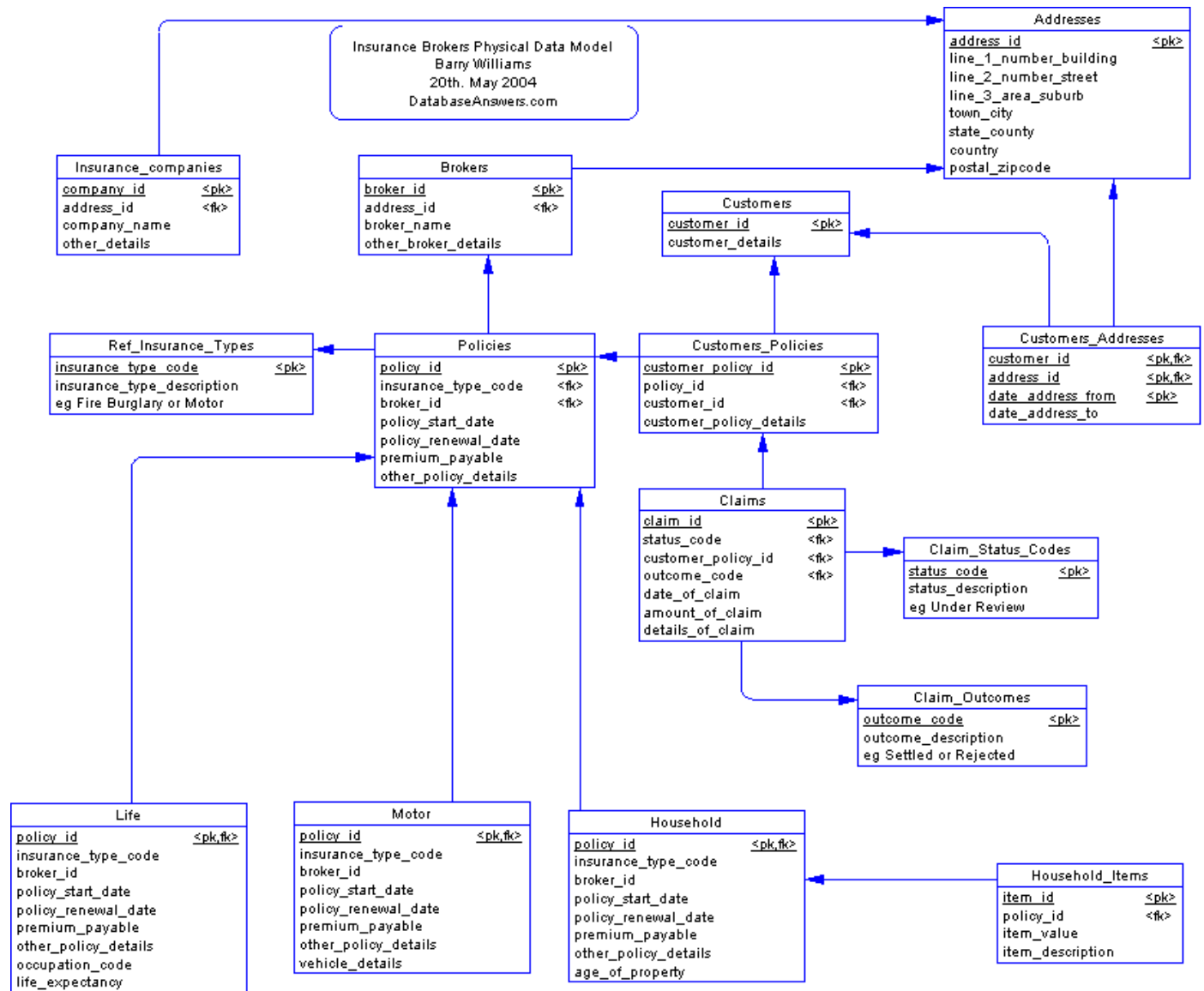
- http://www.databaseanswers.org/data_models/enterprise_data_model_for_insurance/index.htm



15.3.3 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

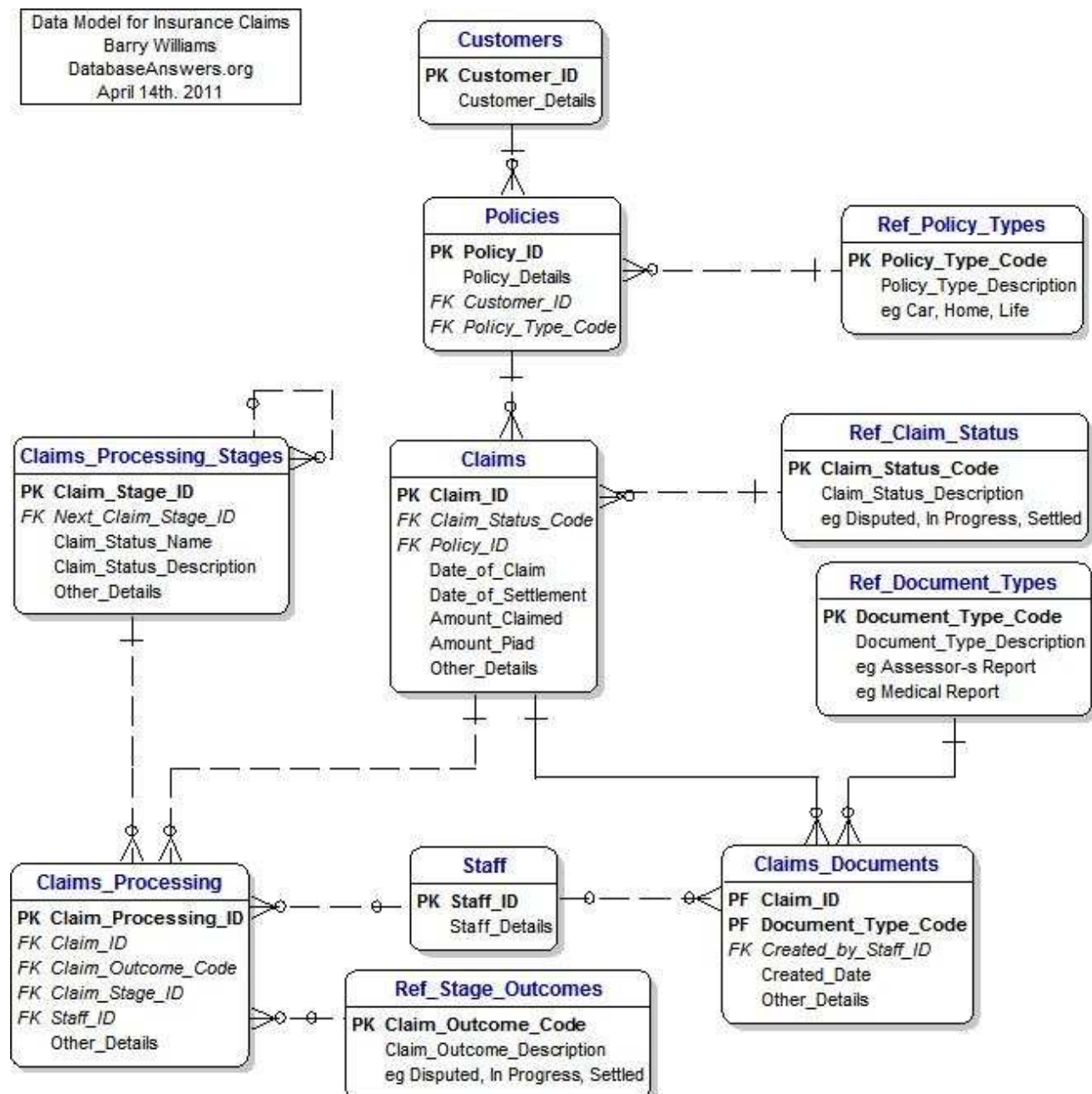


15.3.4 Policies and Claims Subject Area

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

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

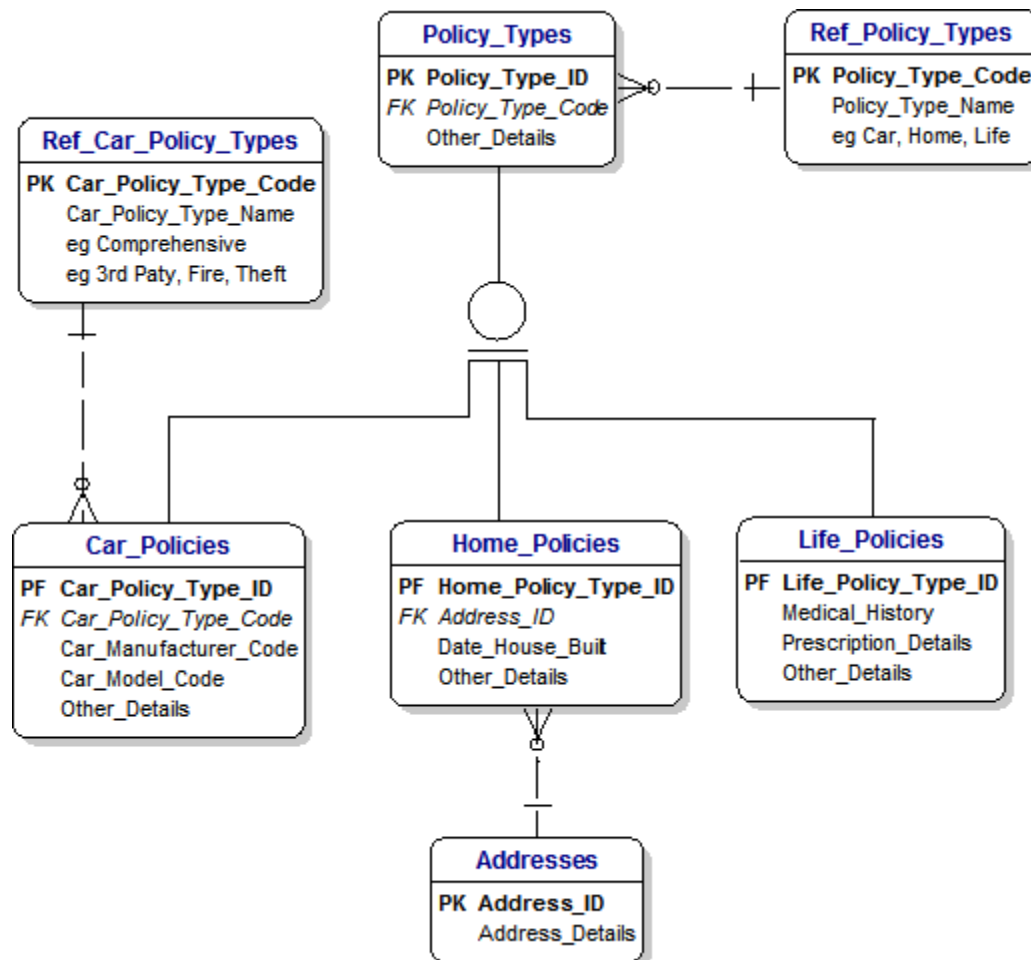
Was: http://www.databaseanswers.org/data_models/insurance_personal/index.htm



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

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

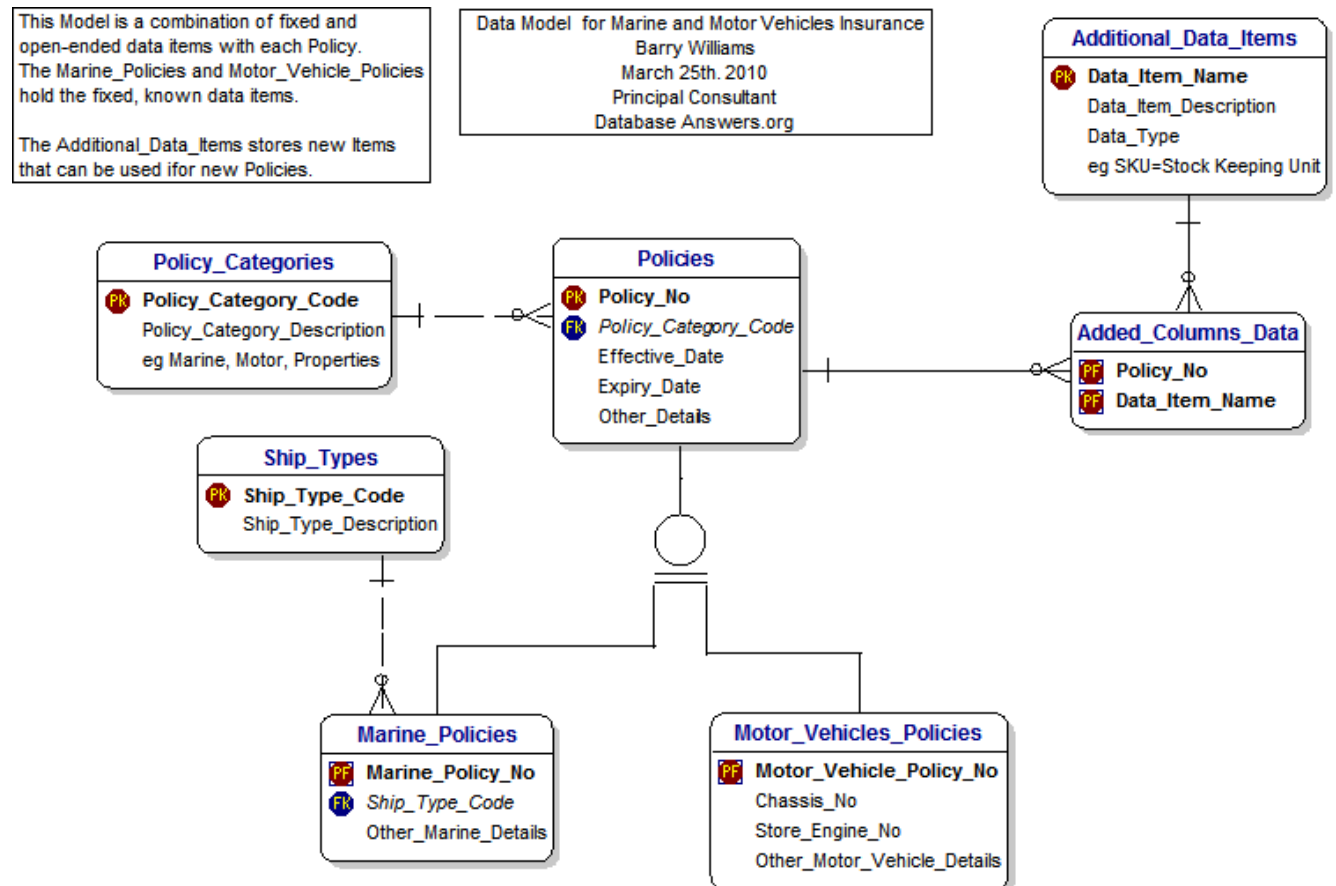
- http://www.databaseanswers.org/data_models/insurance_car_home_life/index.htm



15.3.6 Types of Insurance – Marine and Motor Vehicles Subject Area

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

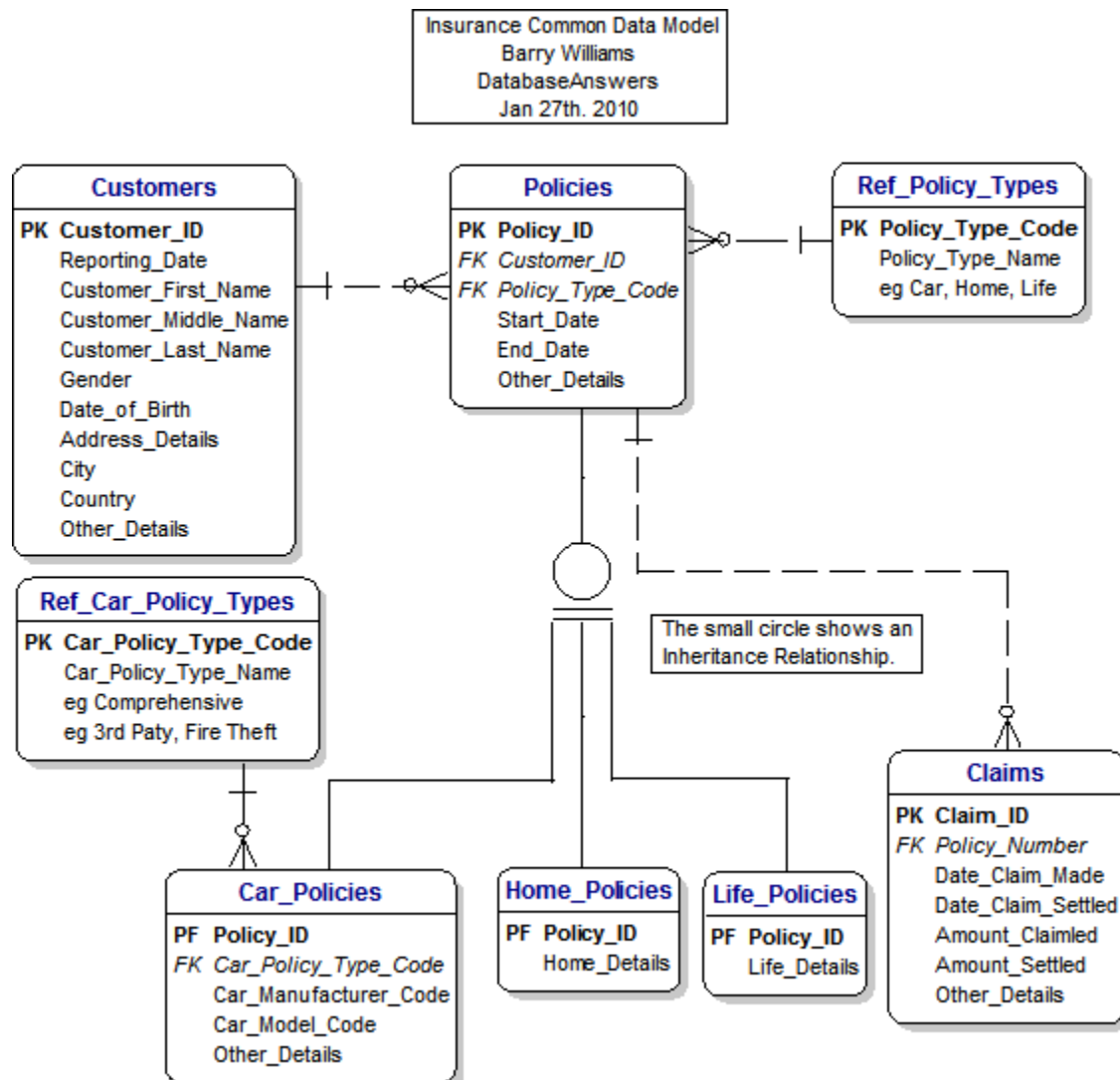
- http://www.databaseanswers.org/data_models/insurance_general/index.htm



15.3.7 Common Data Model

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

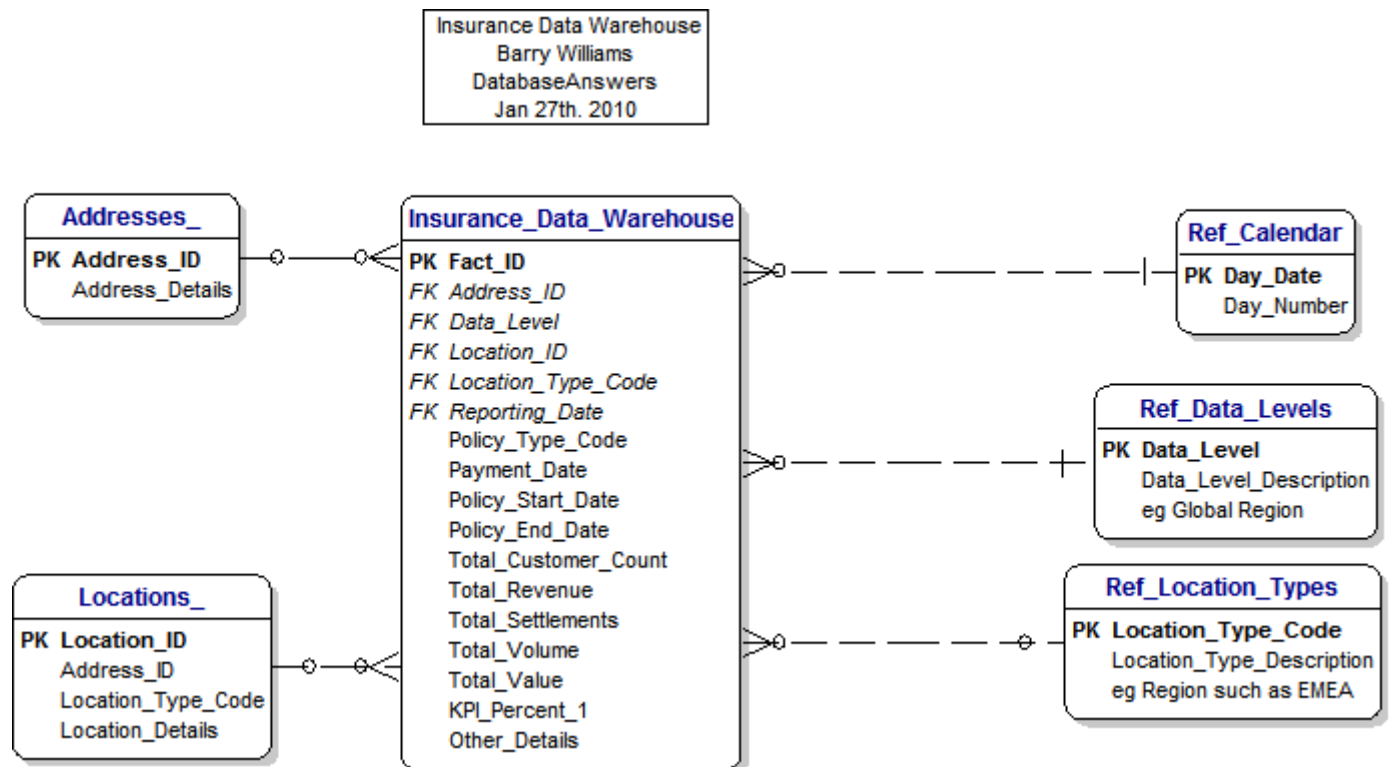
- http://www.databaseanswers.org/data_models/insurance_data_warehouses/common_data_model.htm



15.3.8 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



15.4 Investment Banking

15.4.1 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:

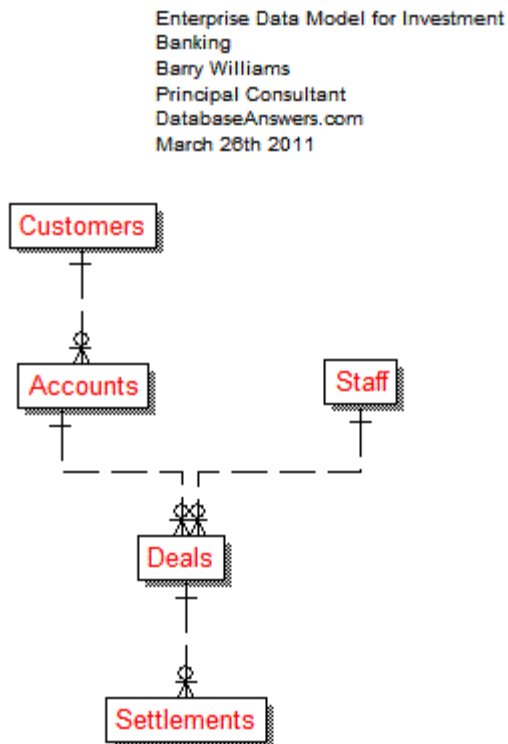
- Accounts
- Clients
- Financial products and services
- Staff

- Settlements
- Trades or deals

15.4.2 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

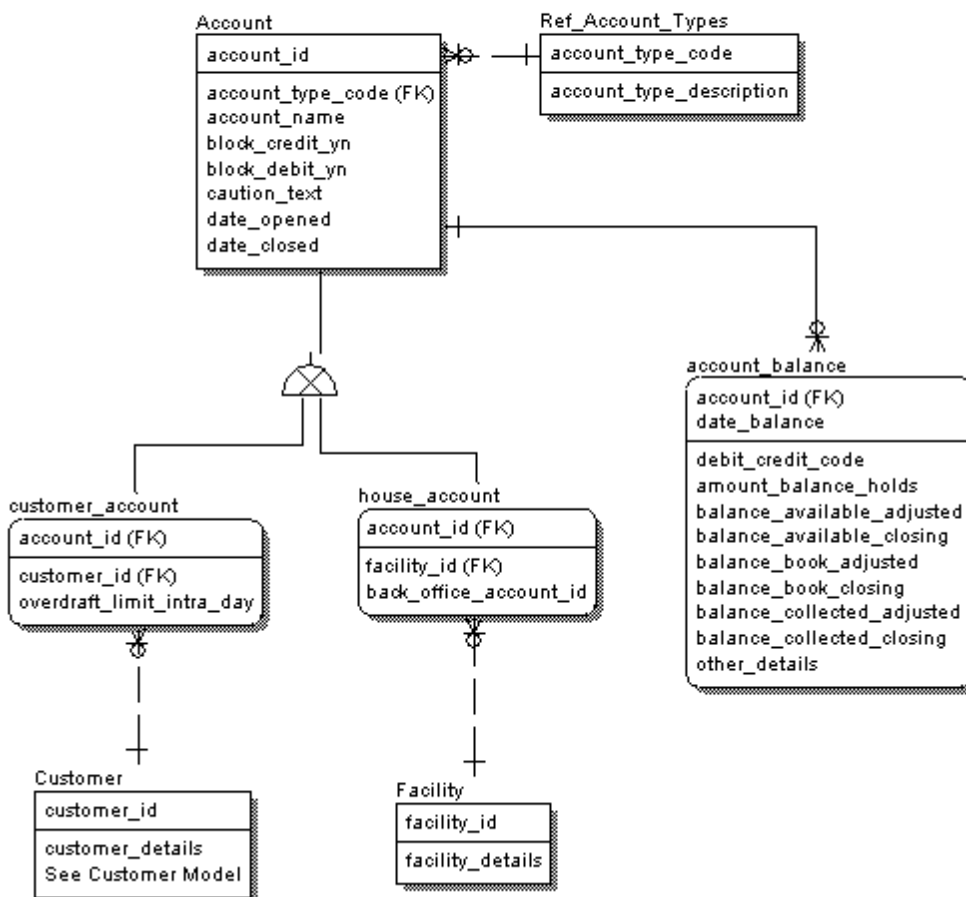


15.4.3 Accounts Subject Area

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

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

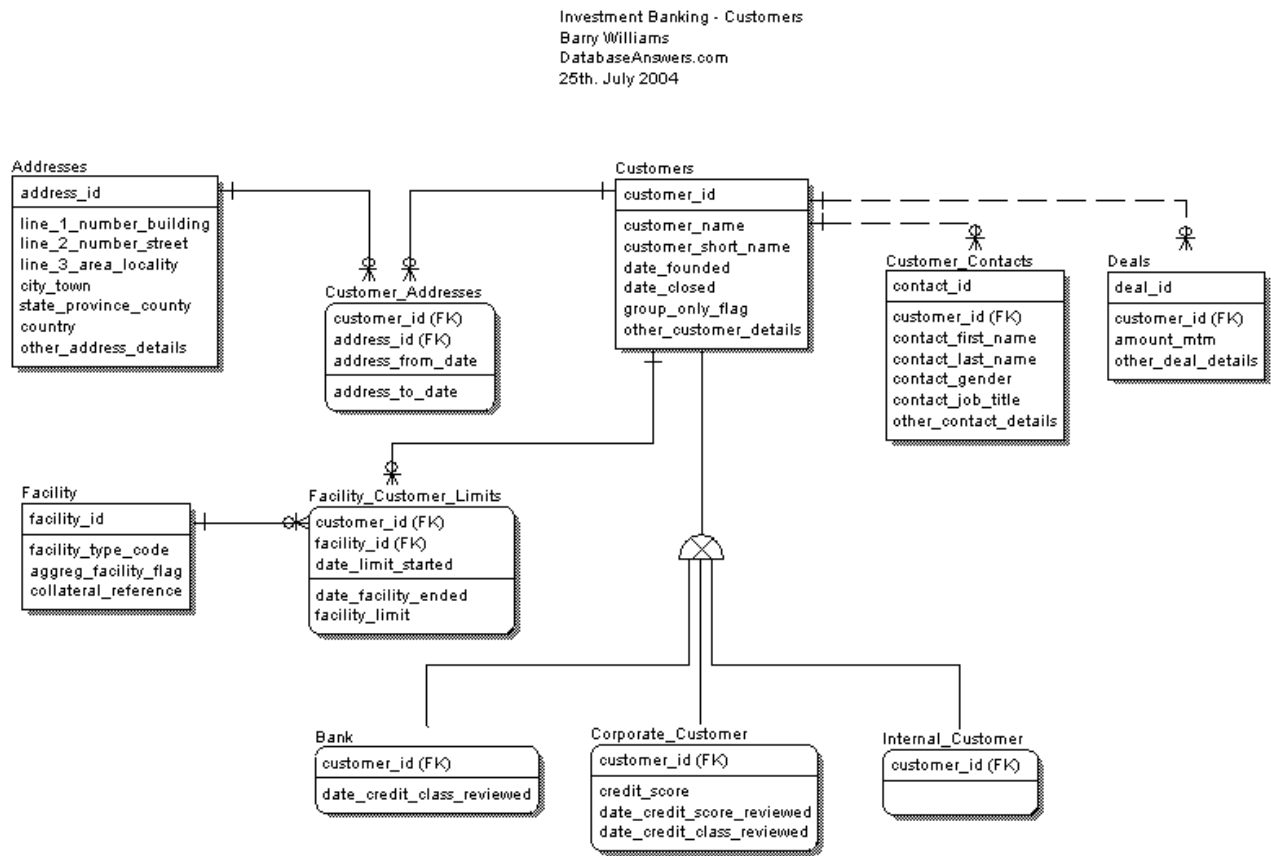
Investment Banking Accounts Model
 Barry Williams
 Principal Consultant
 DatabaseAnswers.com
 23rd. July 2004



15.4.4 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



15.4.5 Deals Subject Area

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

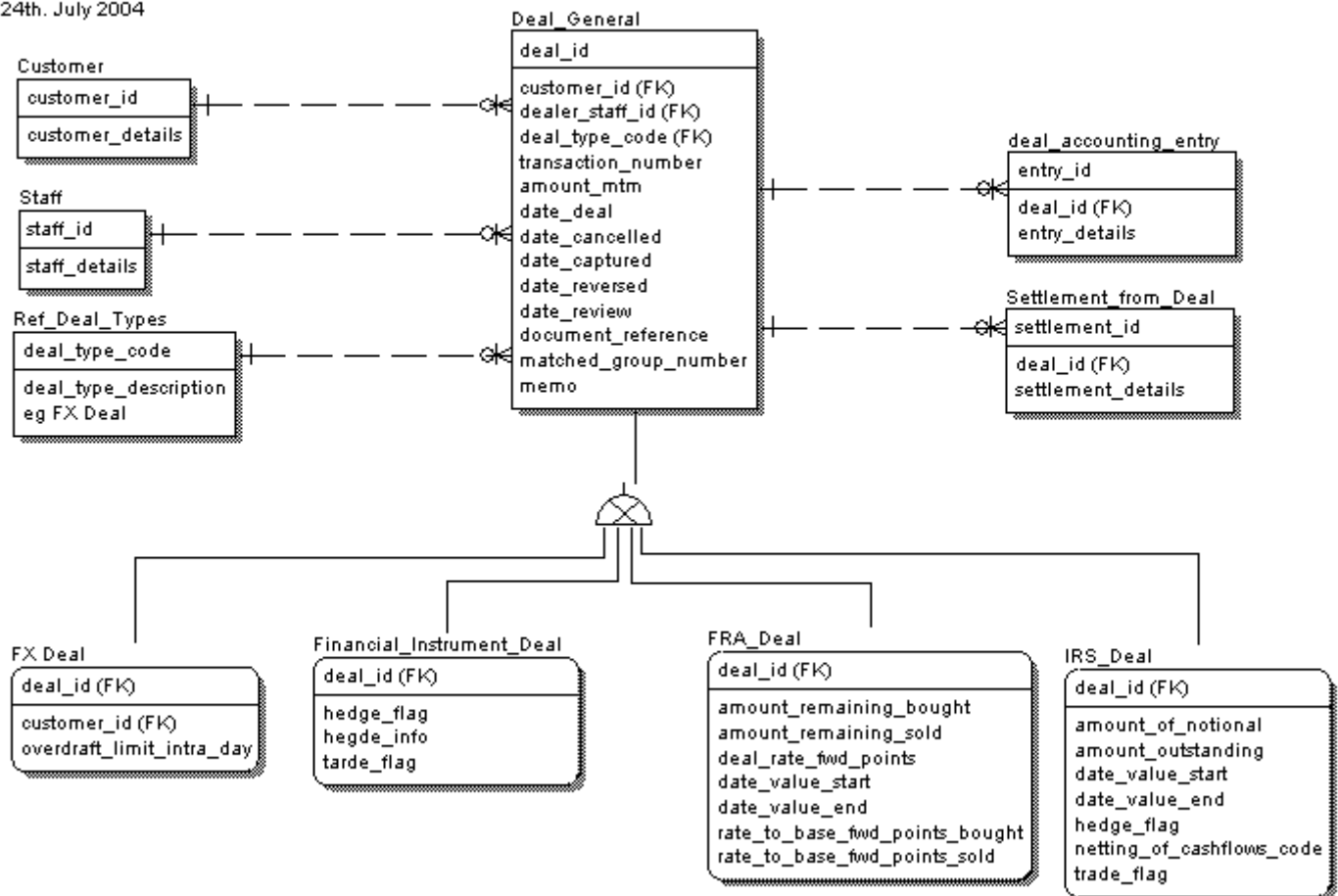
- http://www.databaseanswers.org/data_models/investment_banking/deals_general.htm

Investment Banking - Deals (General)

Barry Williams

DatabaseAnswers.com

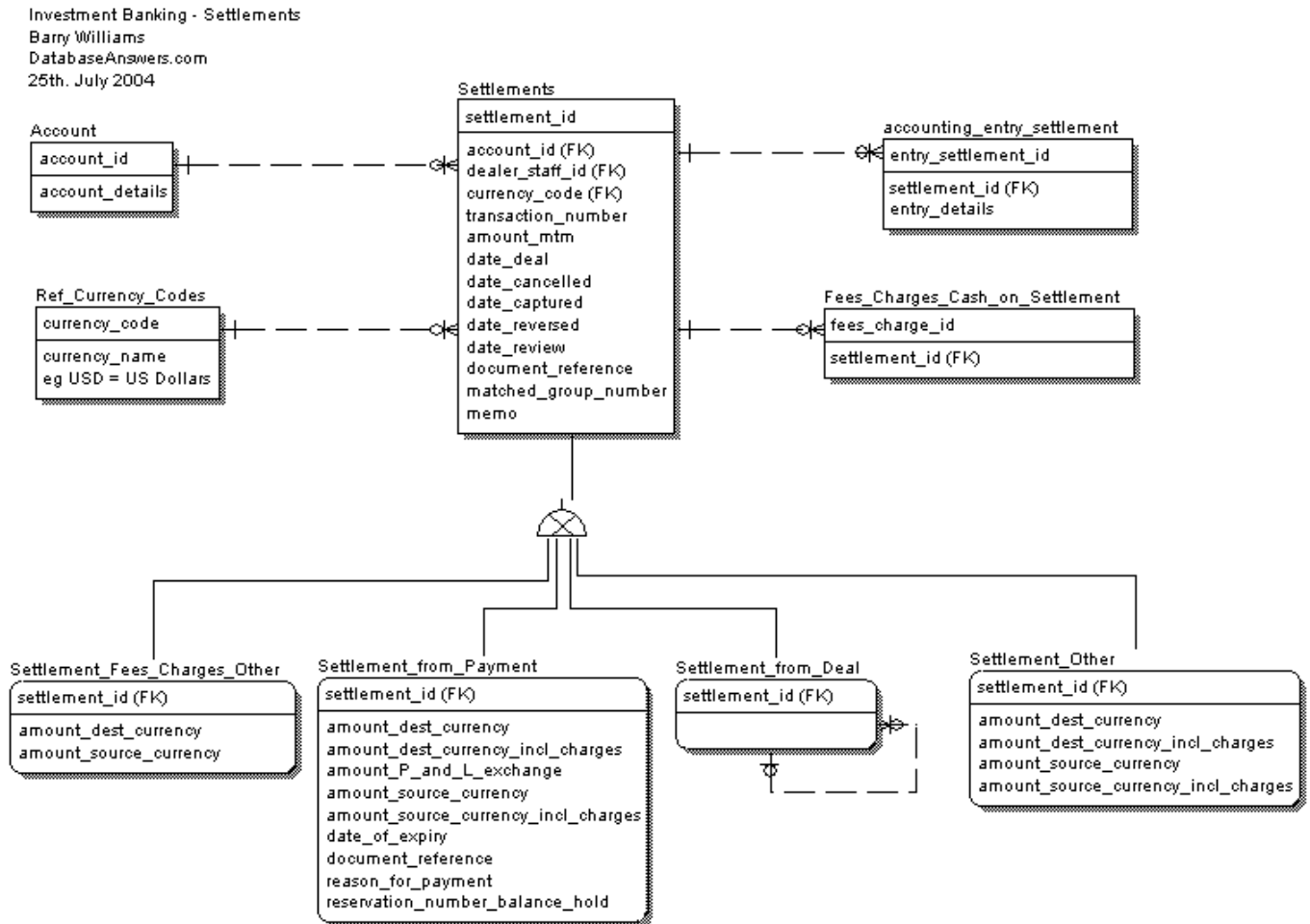
24th. July 2004



15.4.6 Settlements Subject Area

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

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

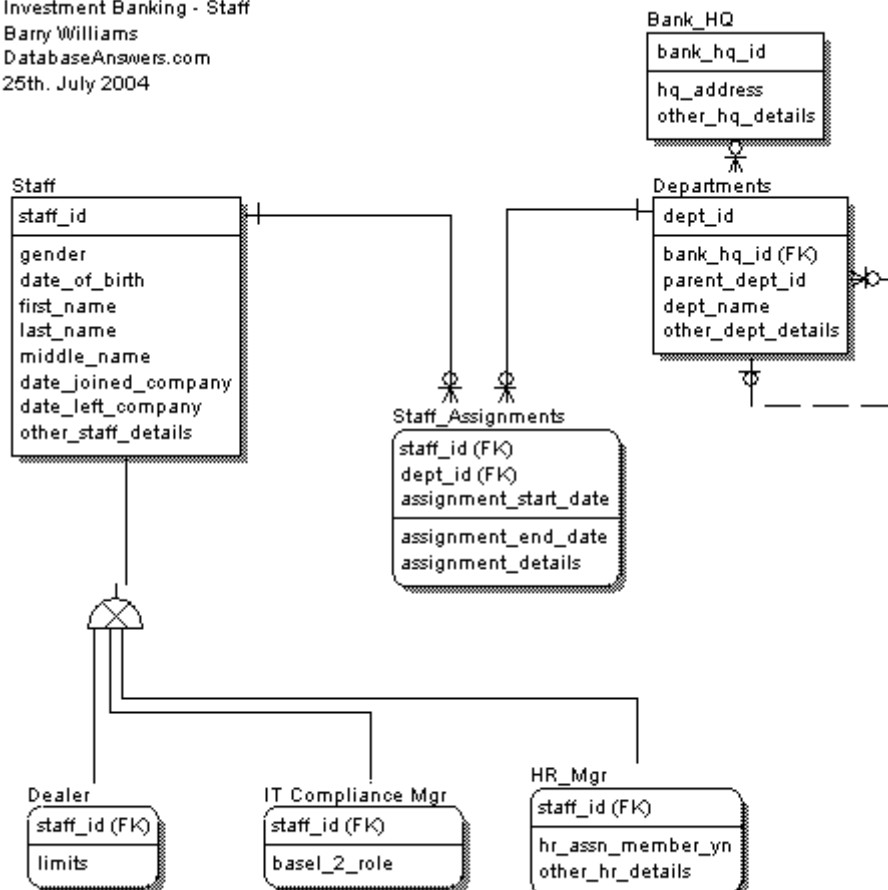


15.4.7 Staff Subject Area

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

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

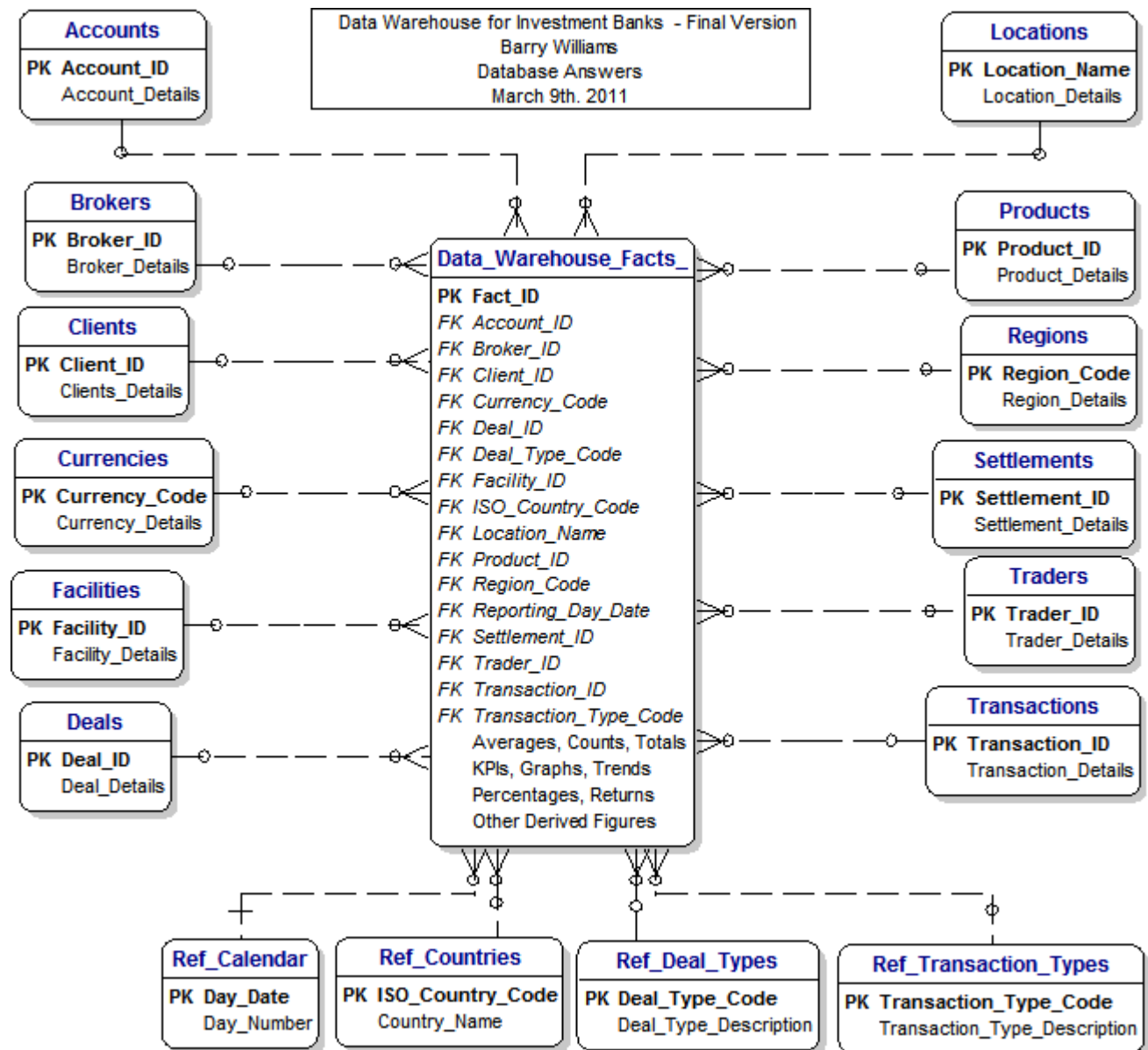
Investment Banking - Staff
Barry Williams
DatabaseAnswers.com
25th. July 2004



15.4.8 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



15.5 Local Government

15.5.1 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."

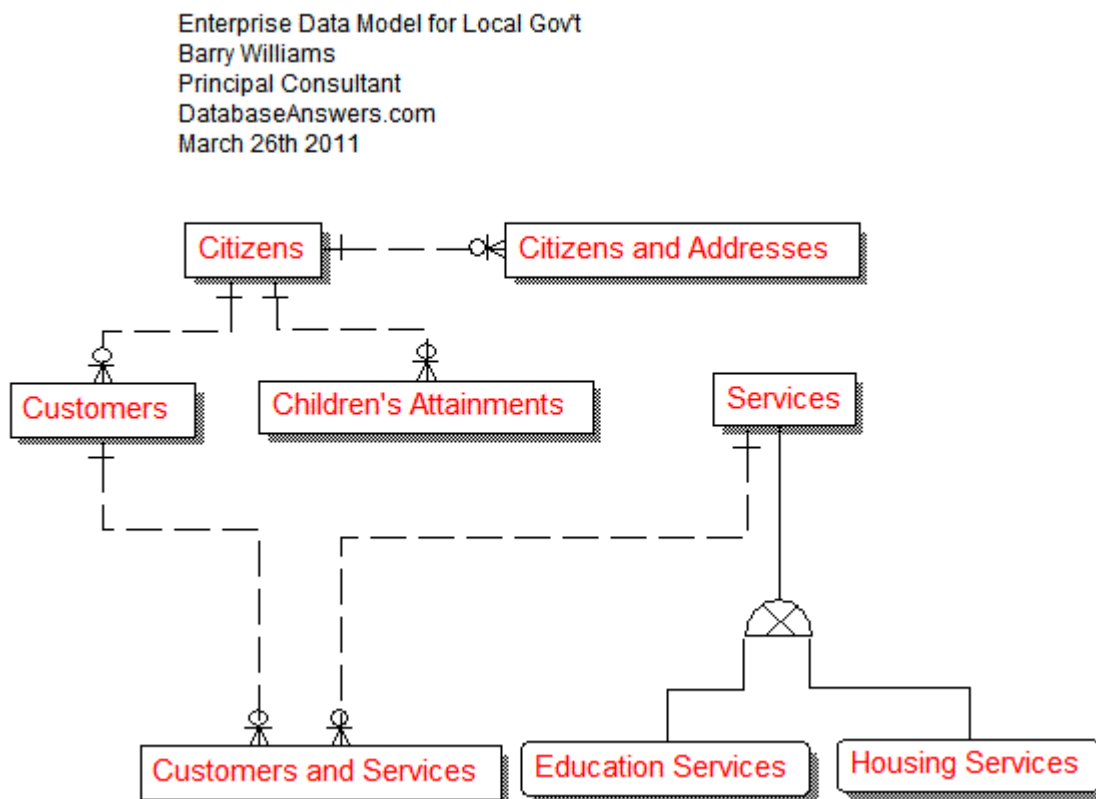
Therefore, the subject areas in the enterprise data model must include:

- Customers or citizens, with both residents and non-residents.
- Services and products
- Service delivery

15.5.2 Enterprise Data Model

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

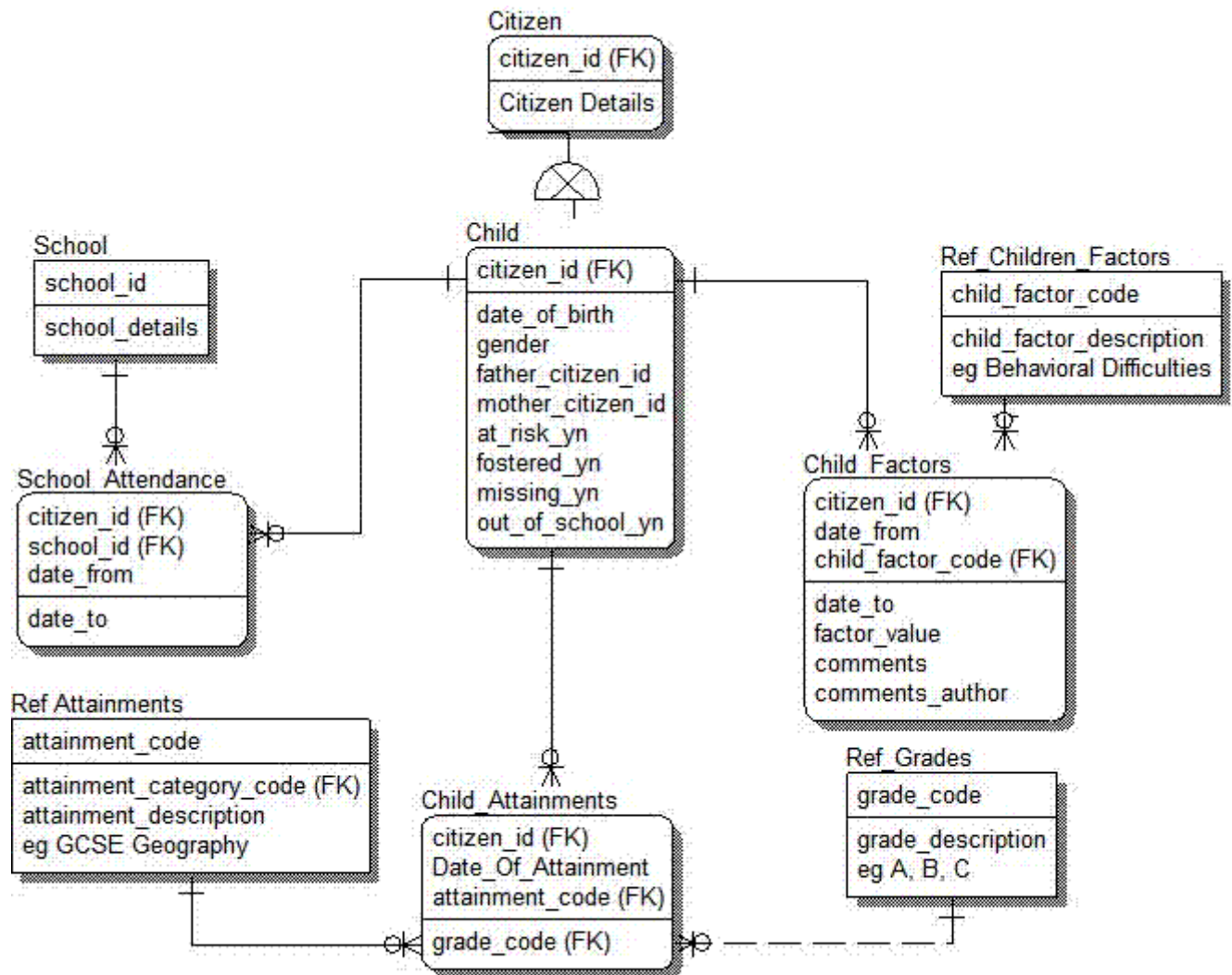
- http://www.databaseanswers.org/data_models/enterprise_data_model_for_local_government/index.htm



15.5.3 Children's Attainments Subject Area

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

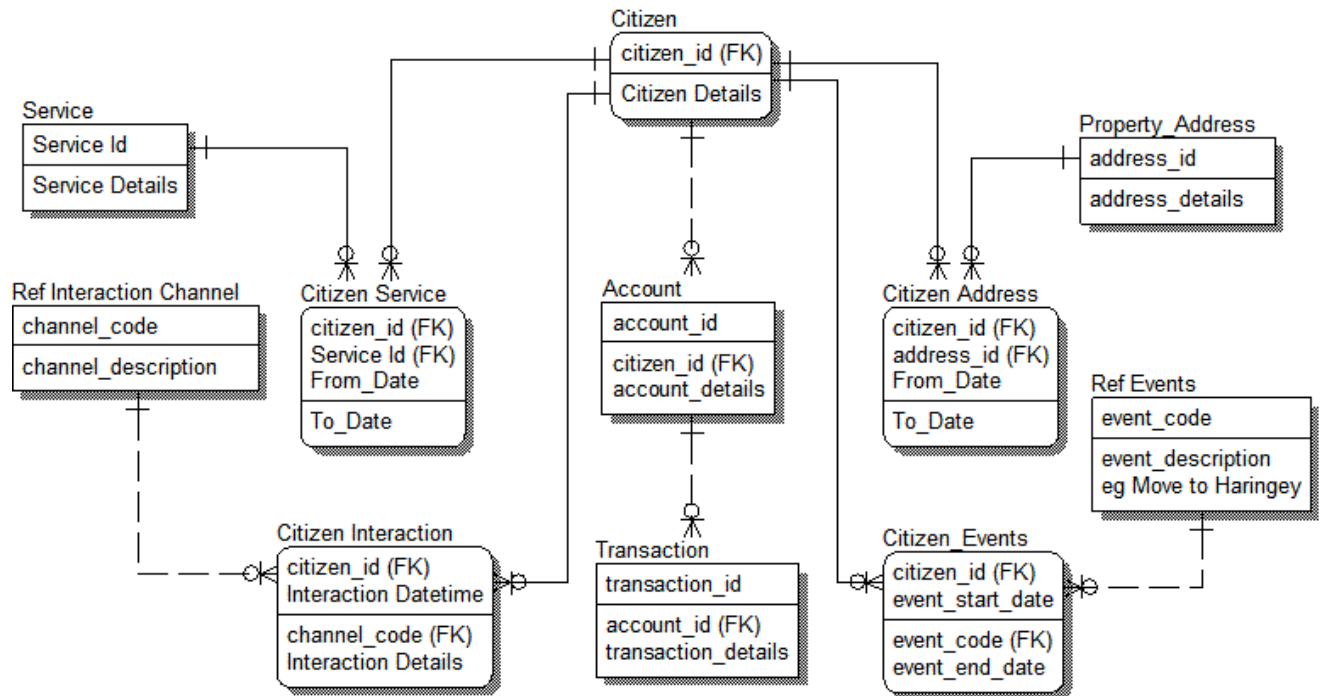
- http://www.databaseanswers.org/data_models/enterprise_data_model_for_local_government/children.htm



15.5.4 Citizens Subject Area

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

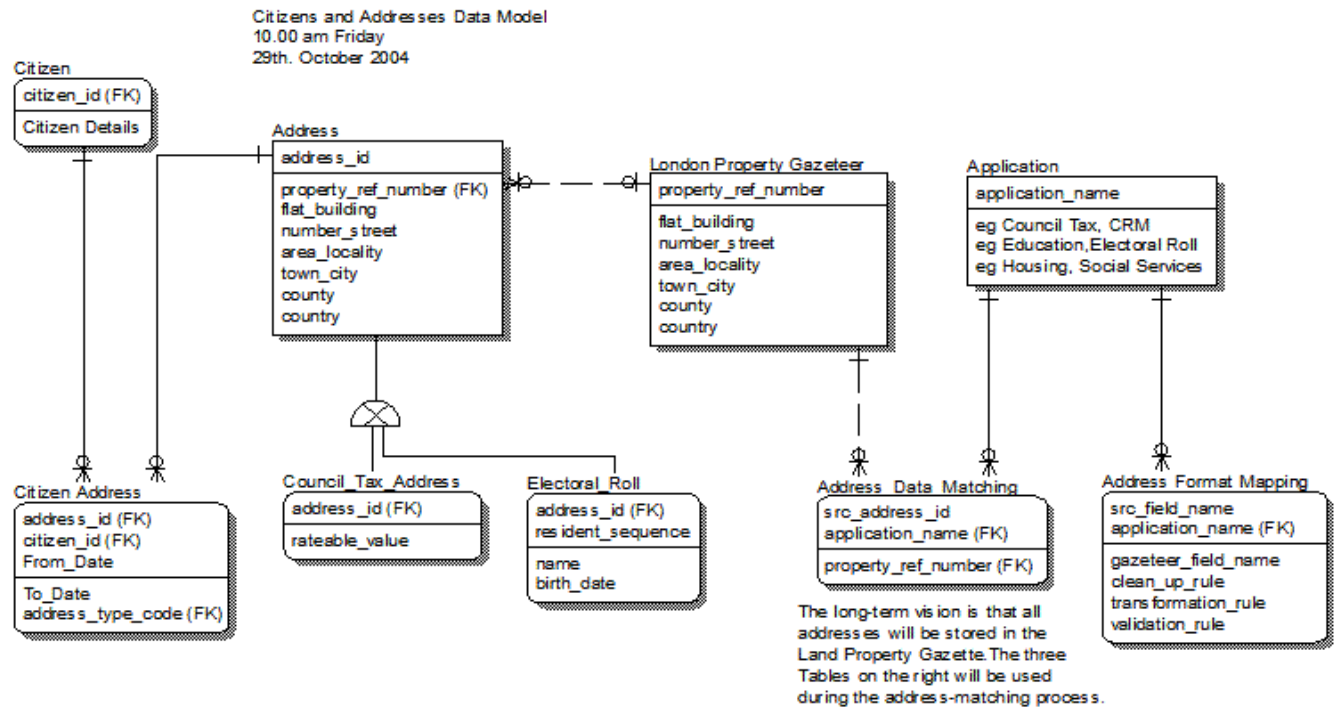
- http://www.databaseanswers.org/data_models/enterprise_data_model_for_local_government/citizens.htm



15.5.5 Citizens and Addresses Subject Area

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

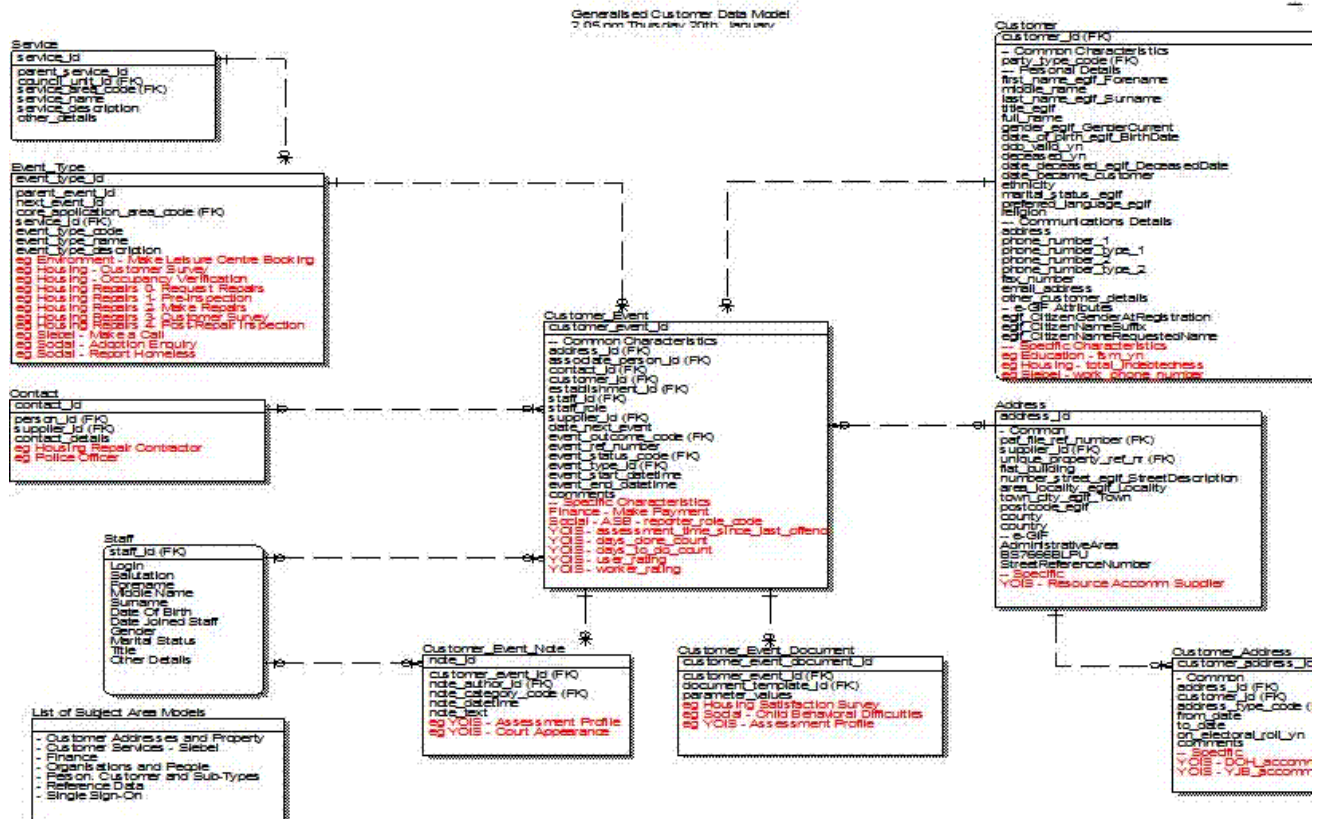
- http://www.databaseanswers.org/data_models/enterprise_data_model_for_local_government/citizens_and_addresses.htm



15.5.6 Citizens and Services Subject Area

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

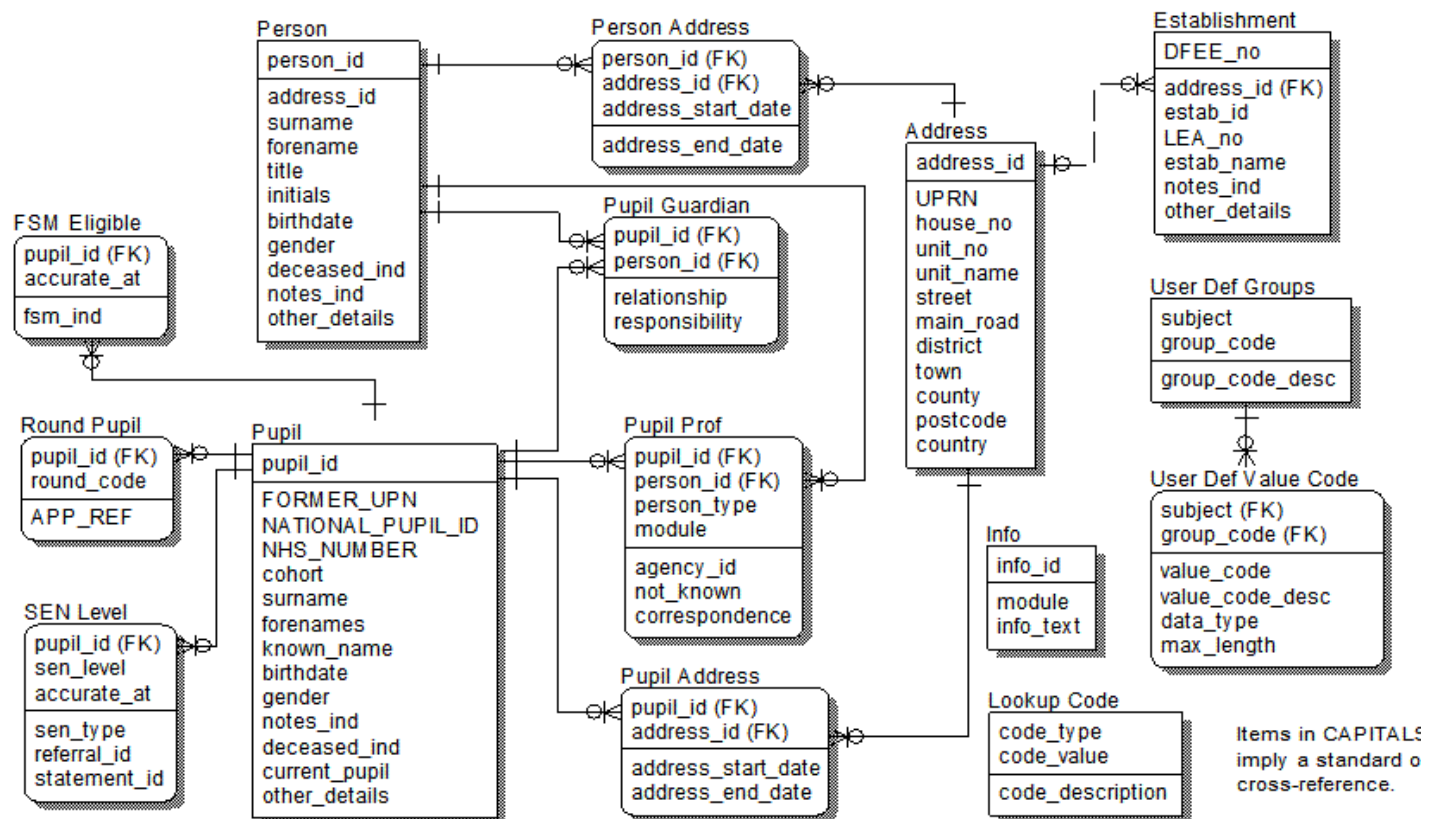
- http://www.databaseanswers.org/data_models/enterprise_data_model_for_local_government/customers_and_services.htm



15.5.7 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_government/education_services.htm

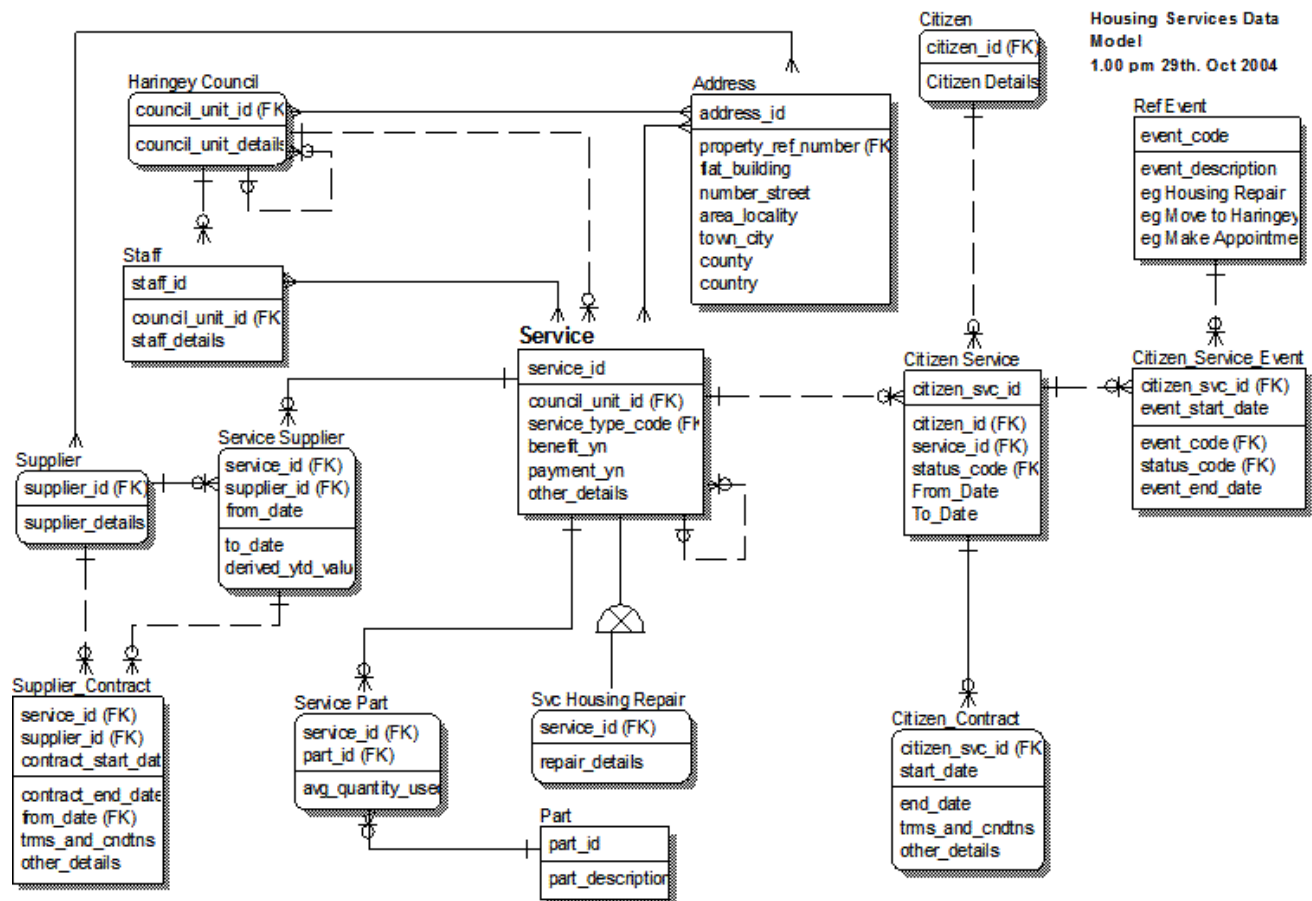


15.5.8 Housing Services Subject Area

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

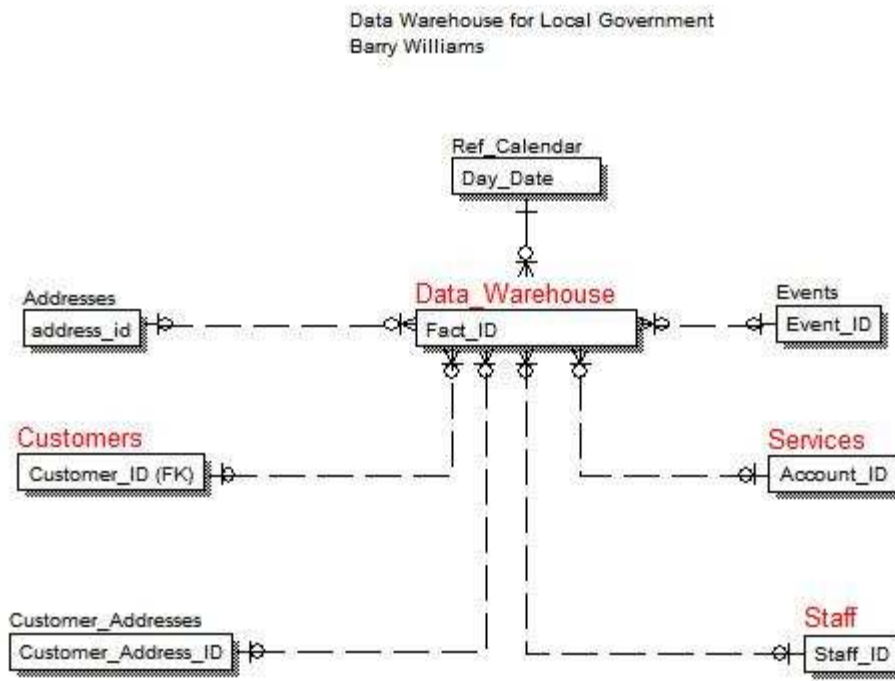
This model includes Haringey Council in London, UK.



15.5.9 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



15.6 Retail

15.6.1 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.

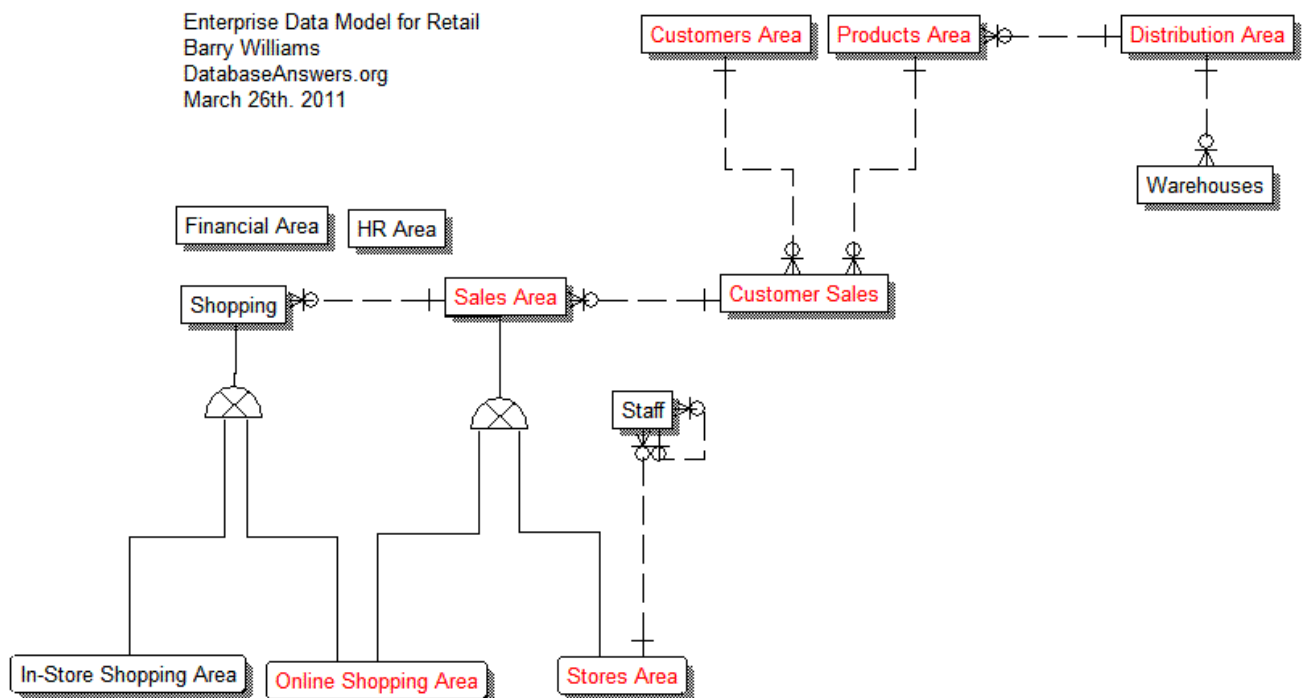
Therefore, the subject areas in the enterprise data model must include:

- Customers
- Distribution
- Inventory
- Marketing
- Products

15.6.2 Enterprise Data Model

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

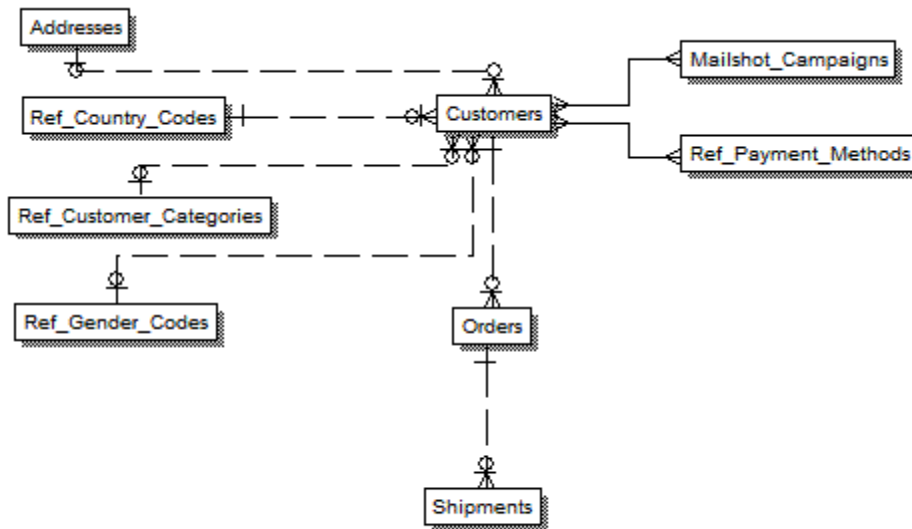
- http://www.databaseanswers.org/data_models/enterprise_data_model_for_retail/index.htm



15.6.3 Customers Subject Area

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

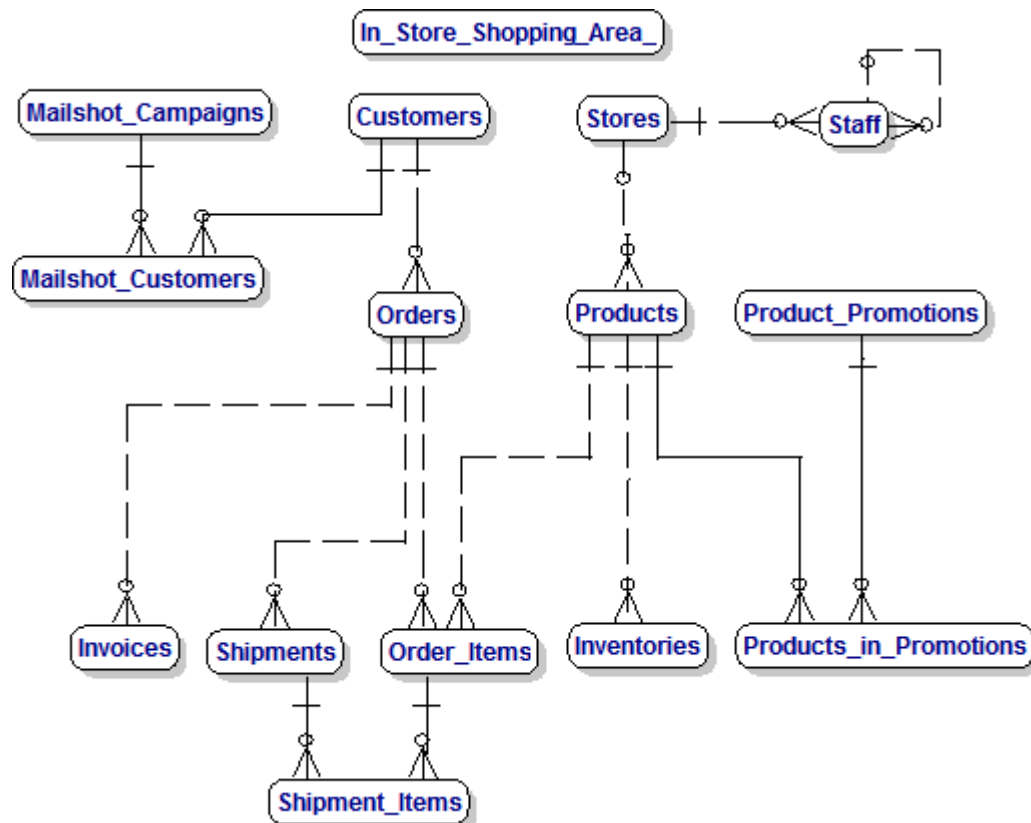
- http://www.databaseanswers.org/data_models/enterprise_data_model_for_retail/customers_area_model.htm



15.6.5 In-Store Shopping Subject Area

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

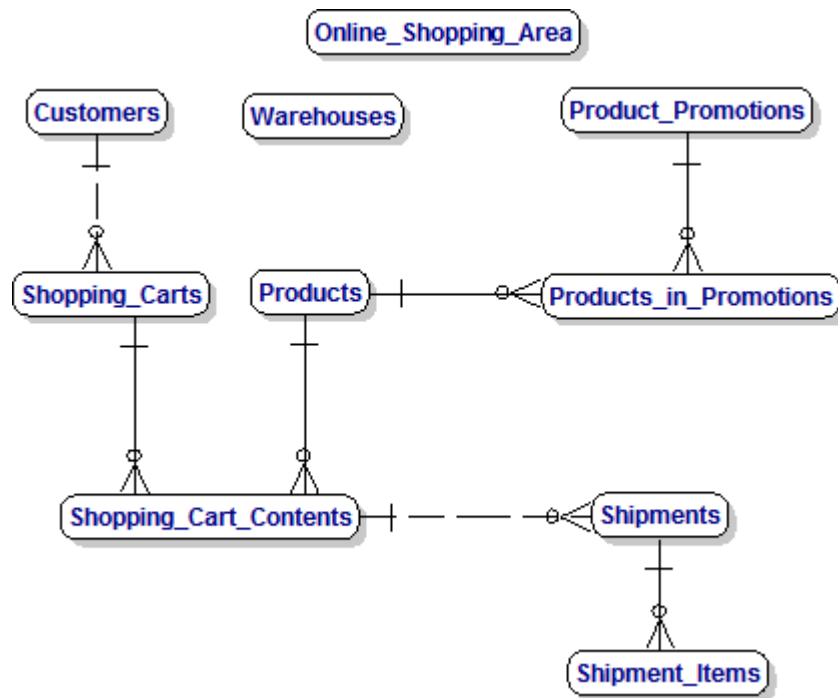
- http://www.databaseanswers.org/data_models/enterprise_data_model_for_retail/in_store_shopping_area_model.htm



15.6.6 Online Shopping Subject Area

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

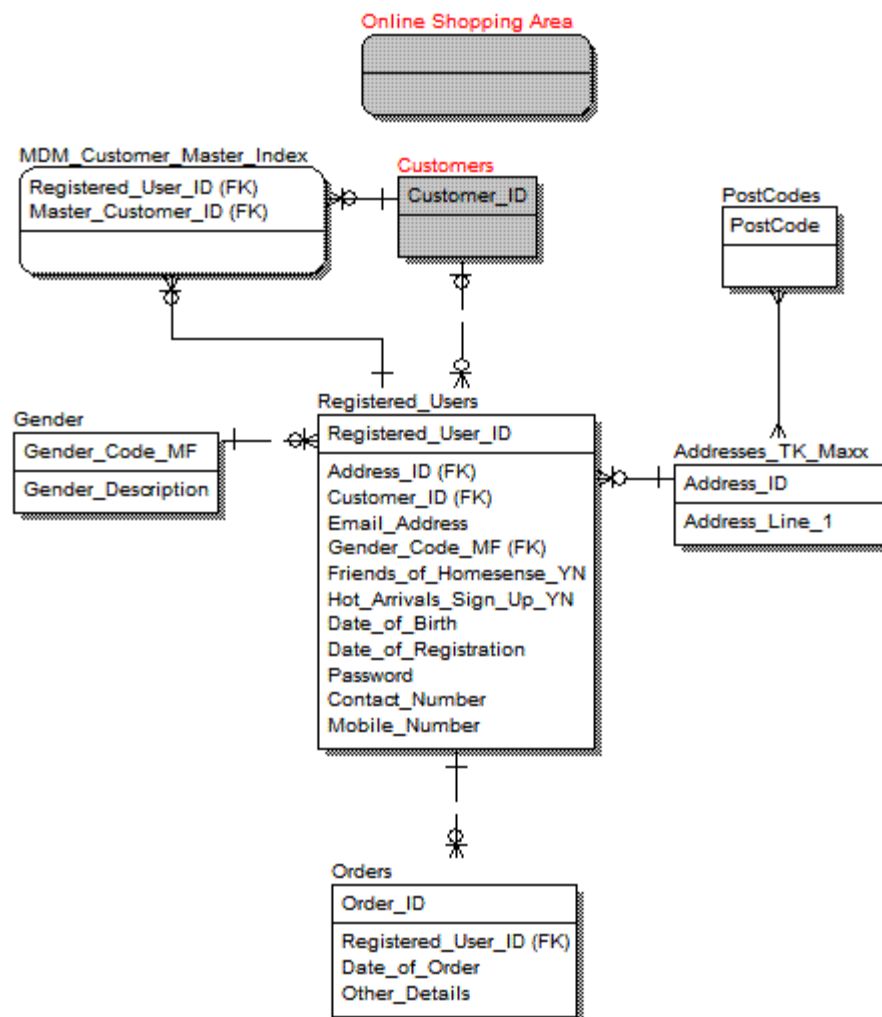
- http://www.databaseanswers.org/data_models/enterprise_data_model_for_retail/online_shopping_area_model.htm



15.6.7 Order History Subject Area

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

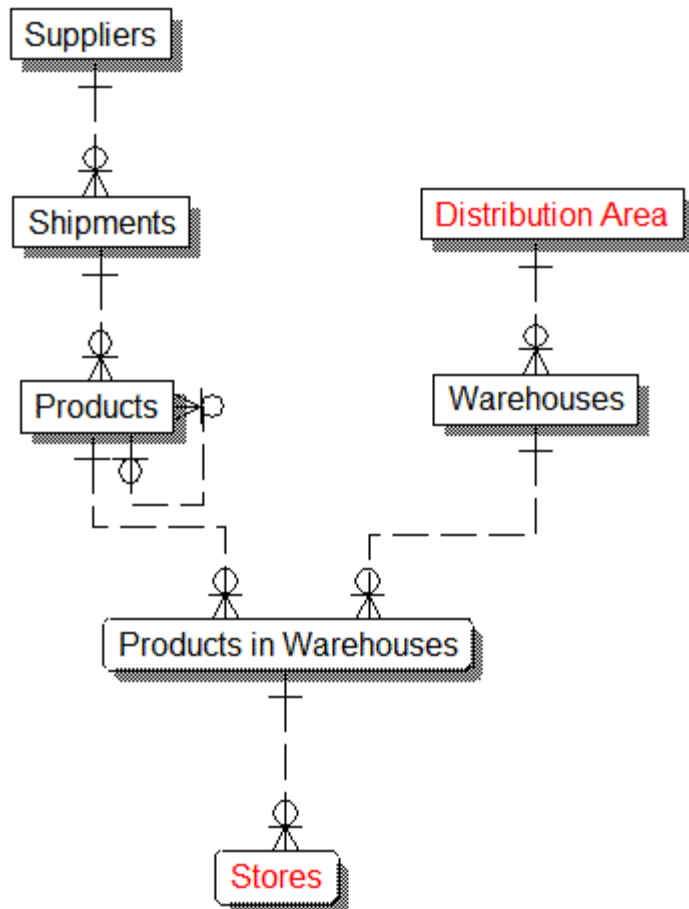
- http://www.databaseanswers.org/data_models/enterprise_data_model_for_retail/Order_History.htm



15.6.8 Products Subject Area

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

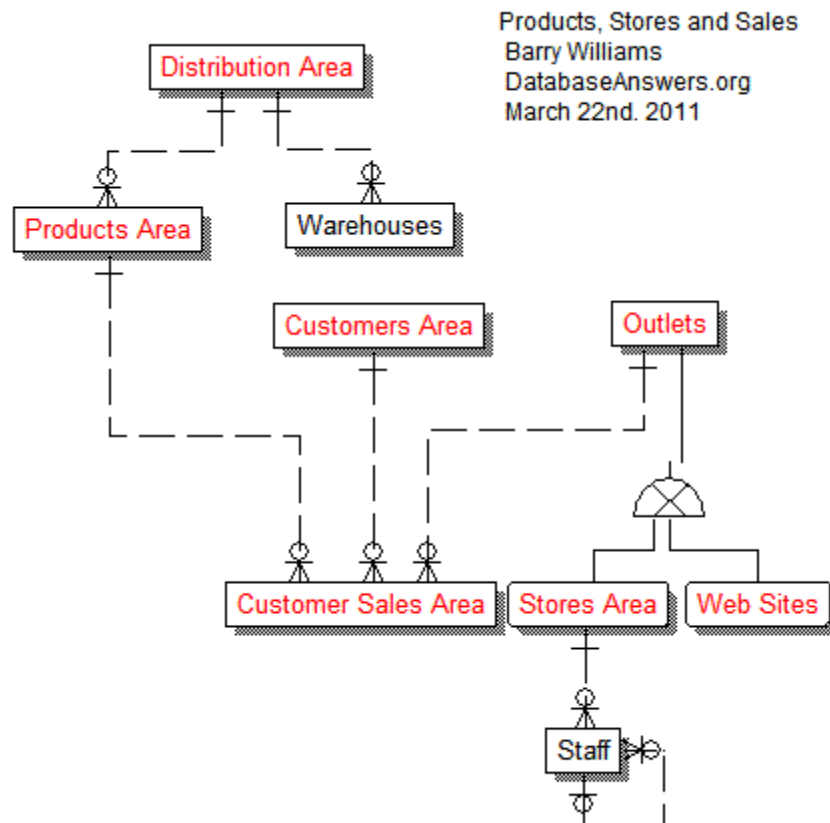
- http://www.databaseanswers.org/data_models/enterprise_data_model_for_retail/Subject_Area_for_Products.htm



15.6.9 Products, Stores and Sales Subject Area

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

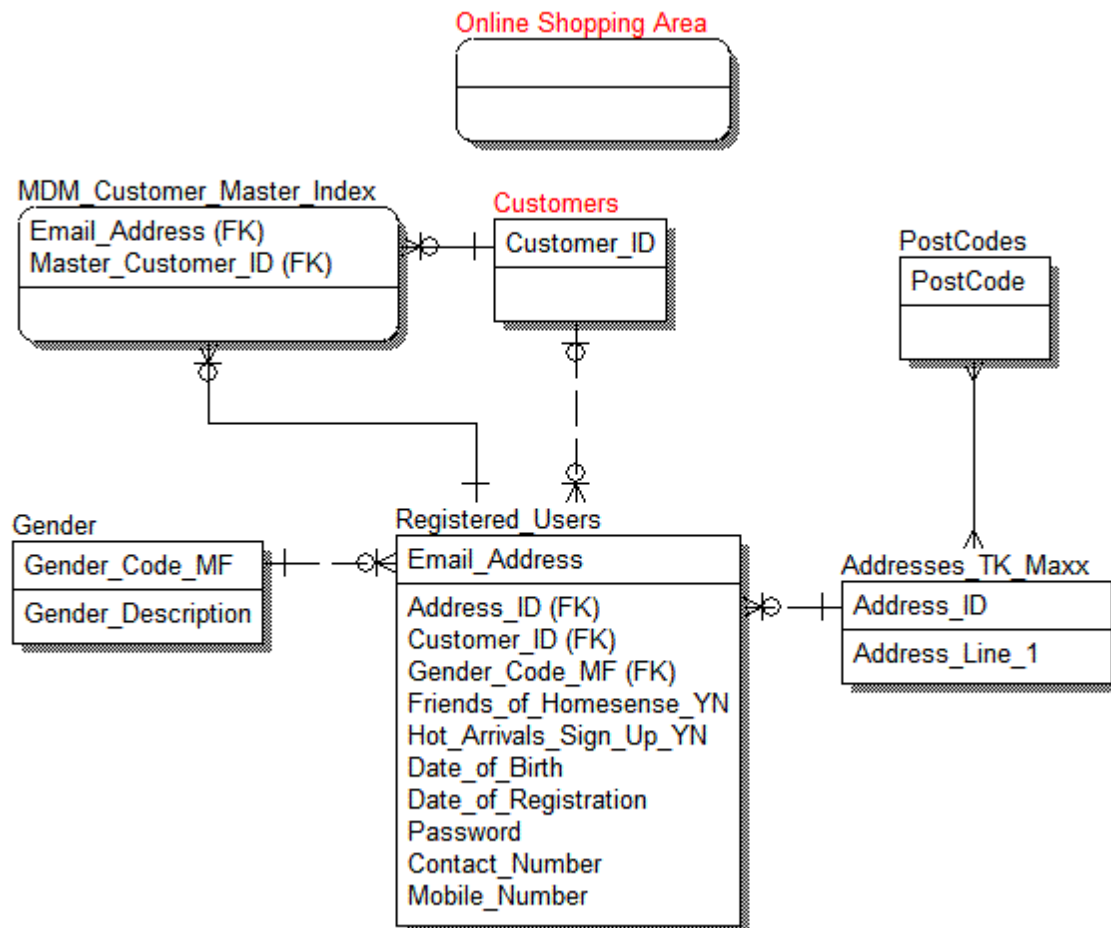
- www.databaseanswers.org/data_models/enterprise_data_model_for_retail/Subject_Area_for_Products_Stores_and_Sales.htm



15.6.10 Registered Users (with MDM) Subject Area

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

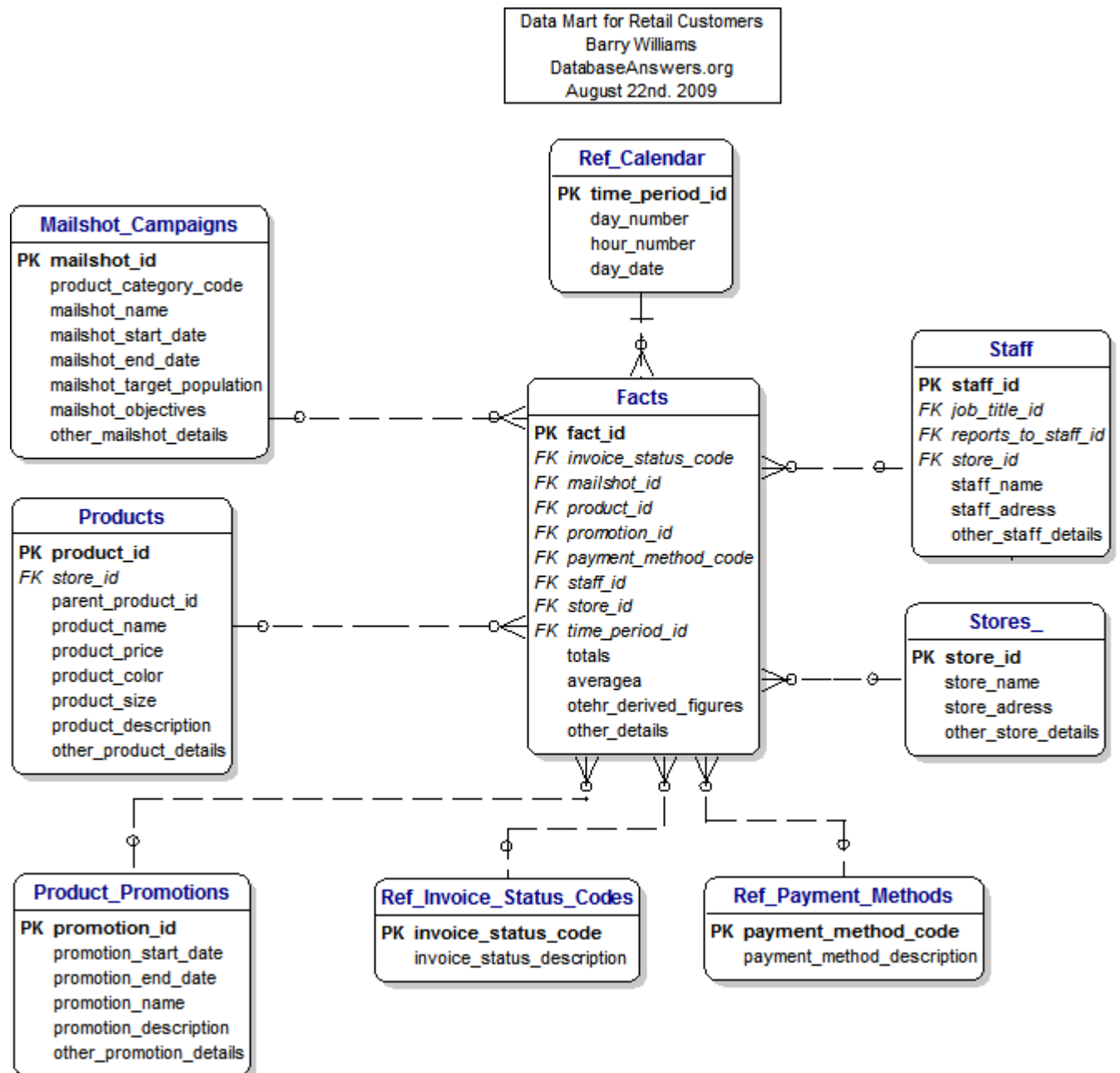
- http://www.databaseanswers.org/data_models/enterprise_data_model_for_retail/Registered_Users.htm



15.6.11 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



15.7 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.

16. Master Data Management (MDM)

16.1 Introduction

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

16.1.1 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.

16.1.2 Why is it Important?

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

16.1.3 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.

16.1.4 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.

16.2 MDM for CRM

This section covers MDM for CRM.

During this short tutorial, we will cover the following topics:

- An approach to MDM
- Creating a data dictionary
- Mapping from sources to the target MDM data model.
- Reference data

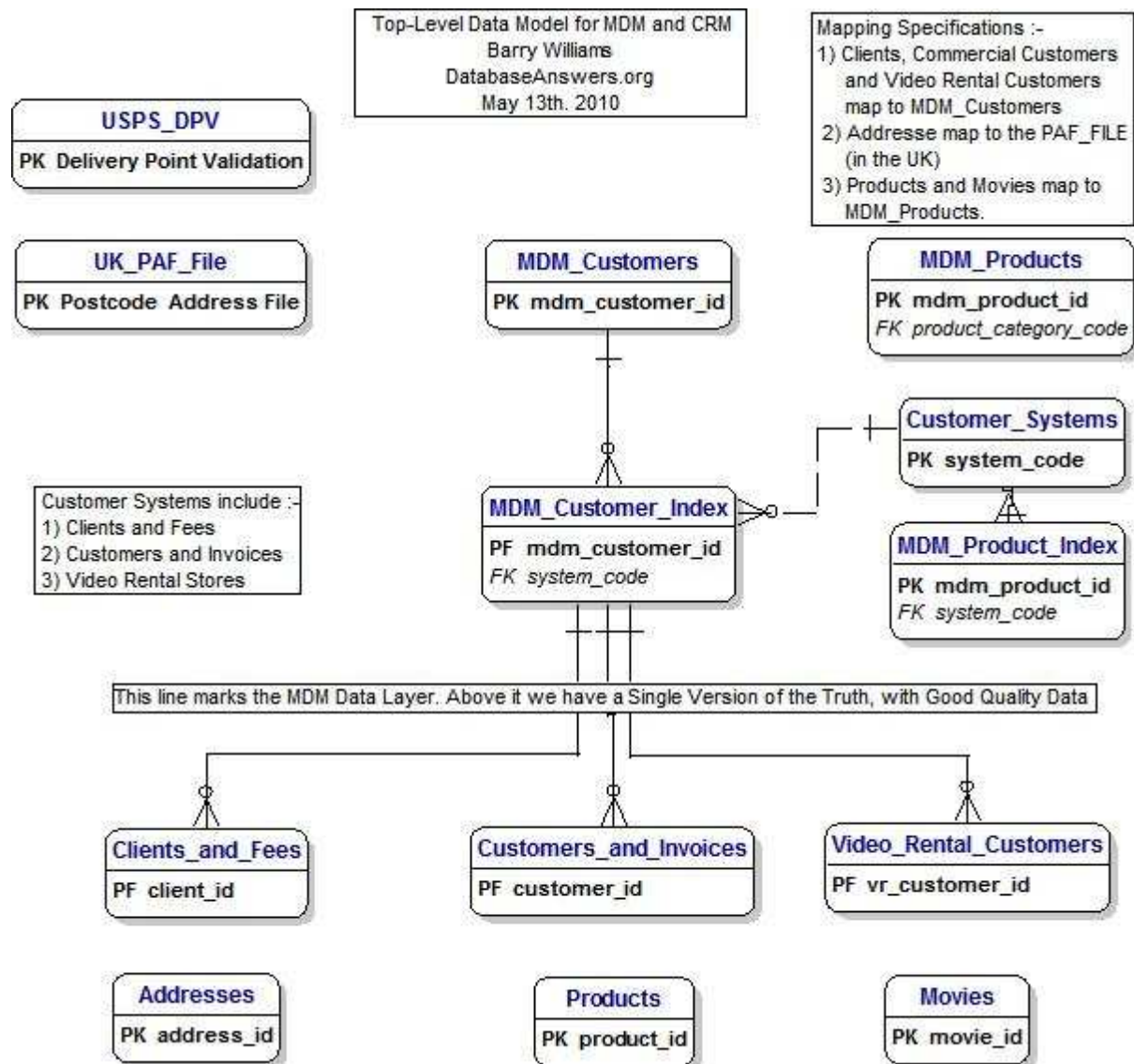
This tutorial presents the steps involved as best practice. Topics covered in this tutorial include data sources and targets, mapping between them, data quality, data governance and reference data.

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.

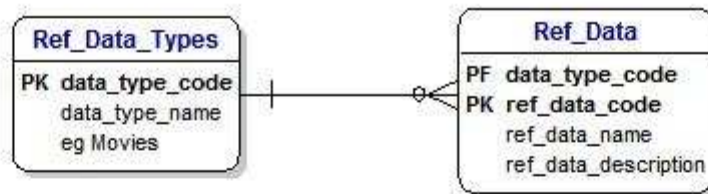
16.2.1 Top-Level CRM Data Model

This is the MDM CRM data model that is the target for data migration.



16.2.2 Reference Data Model

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

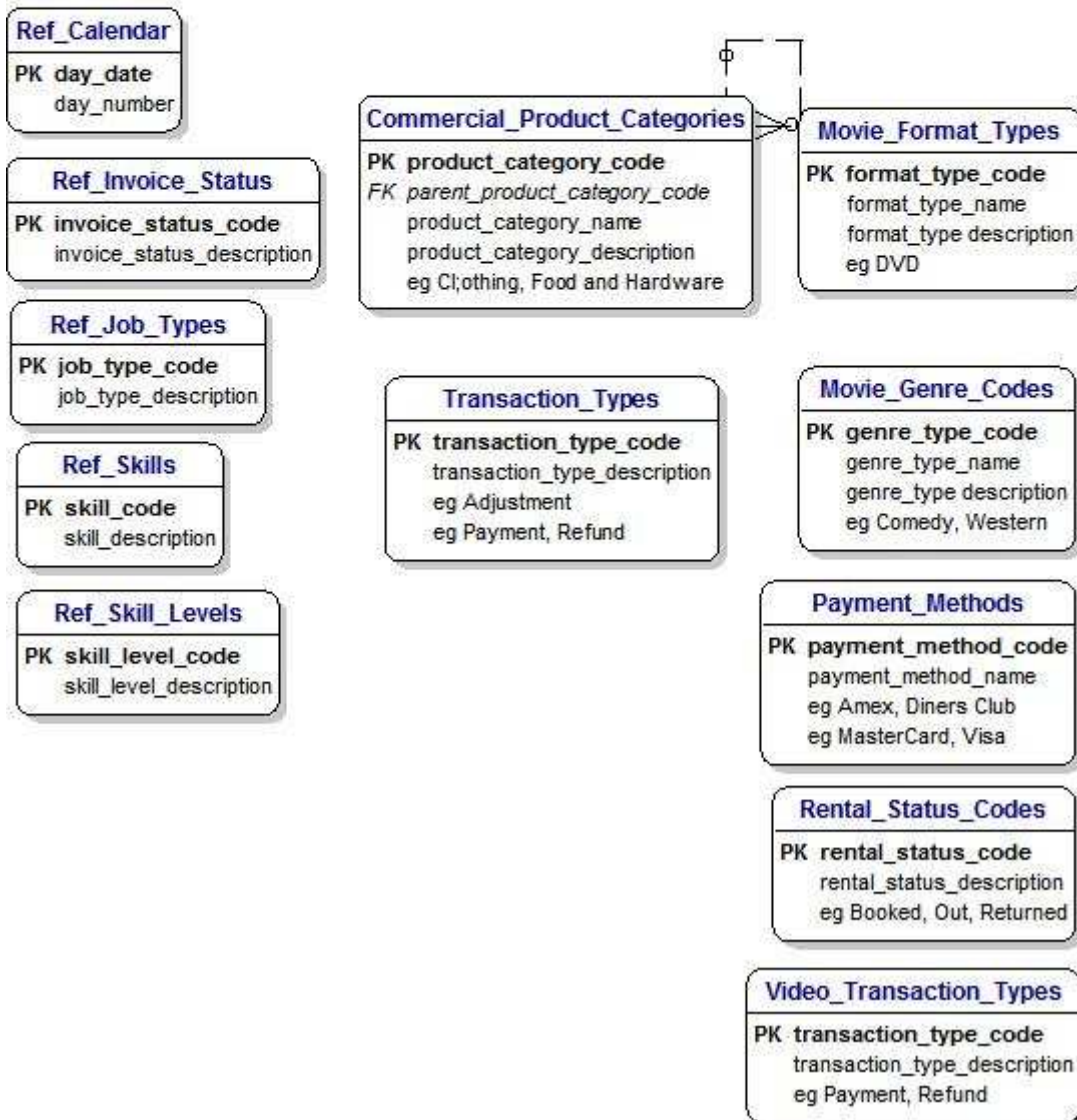


This line marks the MDM Data Layer. Above it we have a Single Version of the Truth, with Good Quality Data

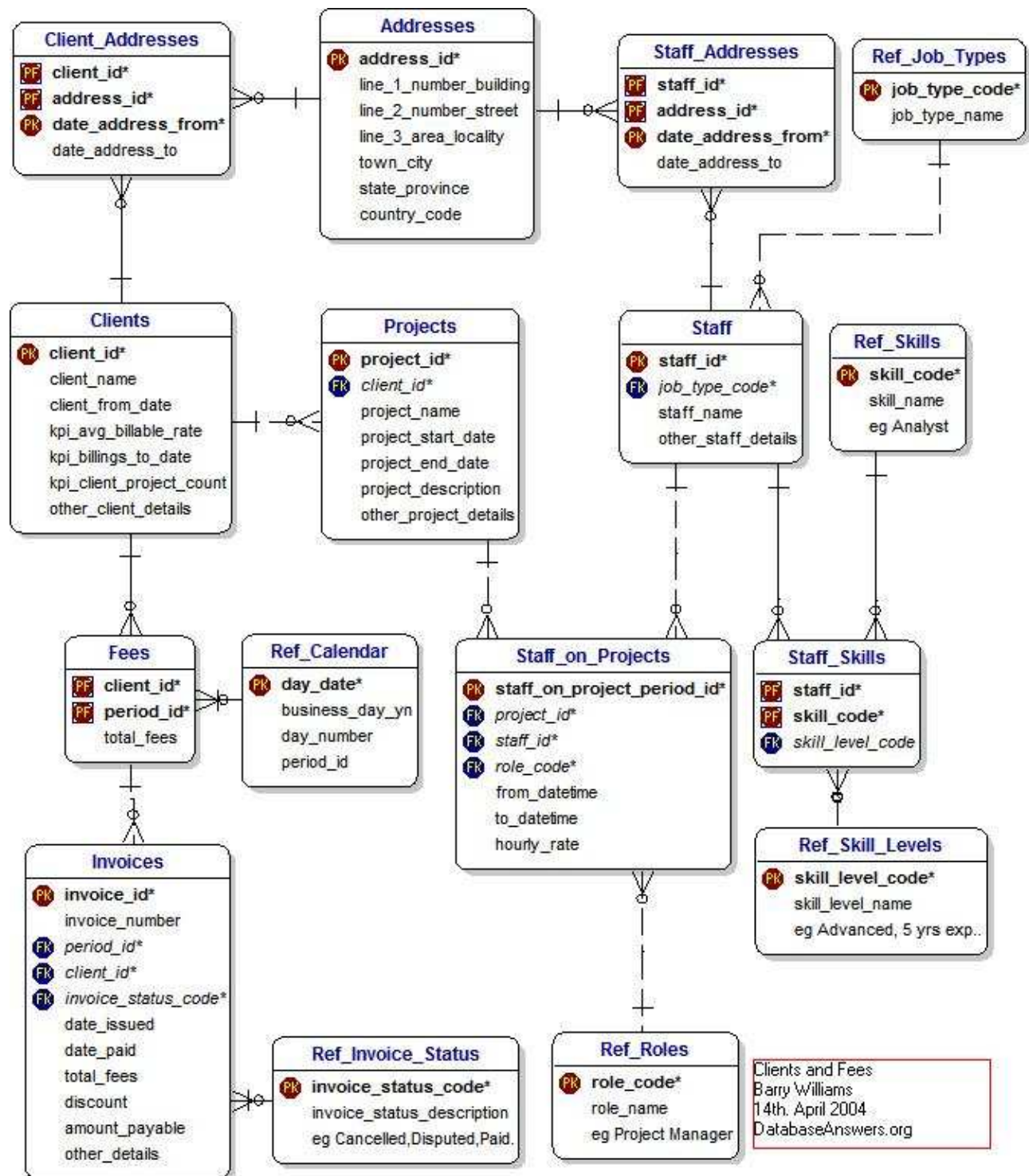
Ref Data for Clients and Fees

Ref Data for Customers and Invoices

Ref Data for Video Rentals

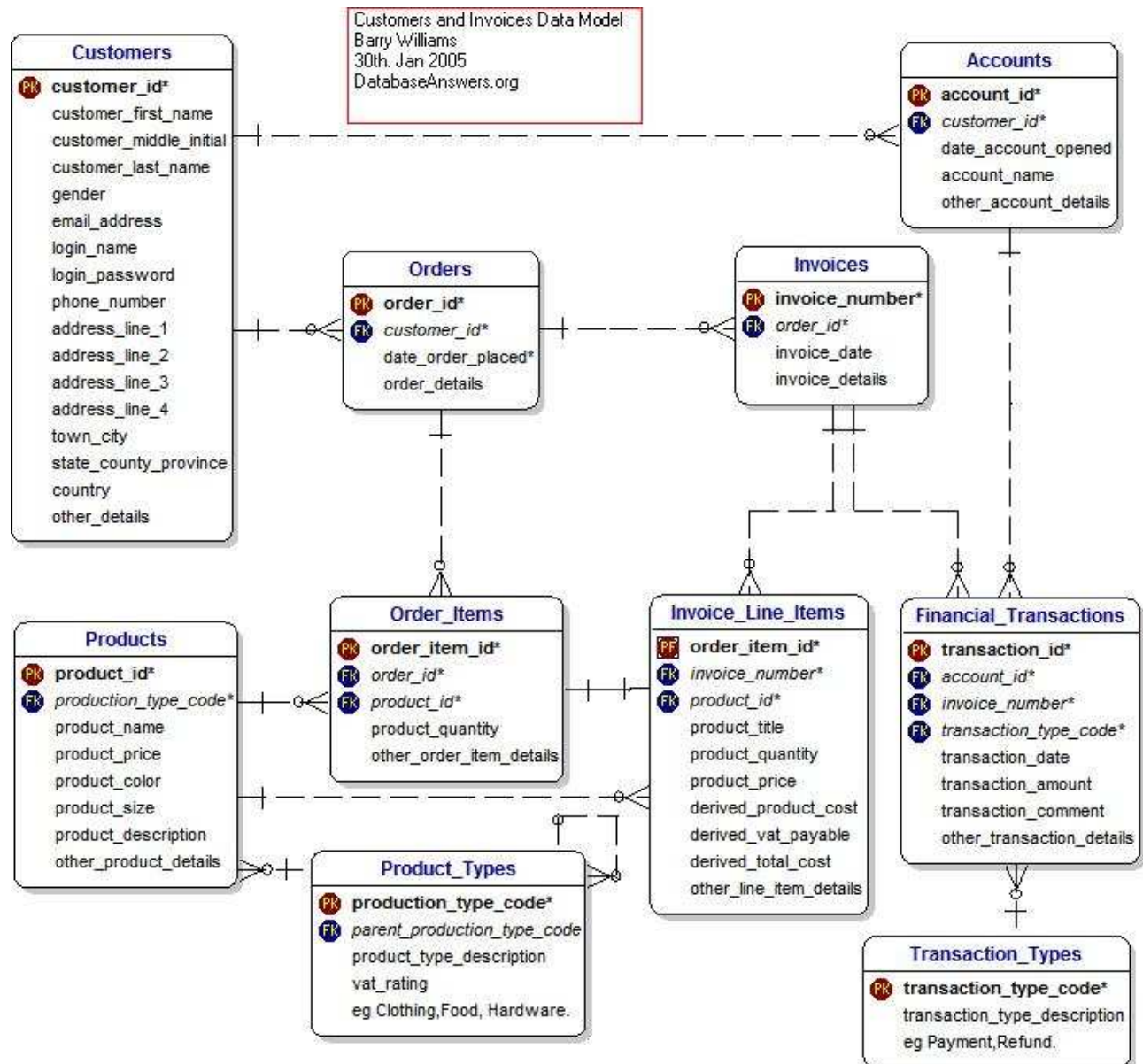


16.2.3 Clients and Fees

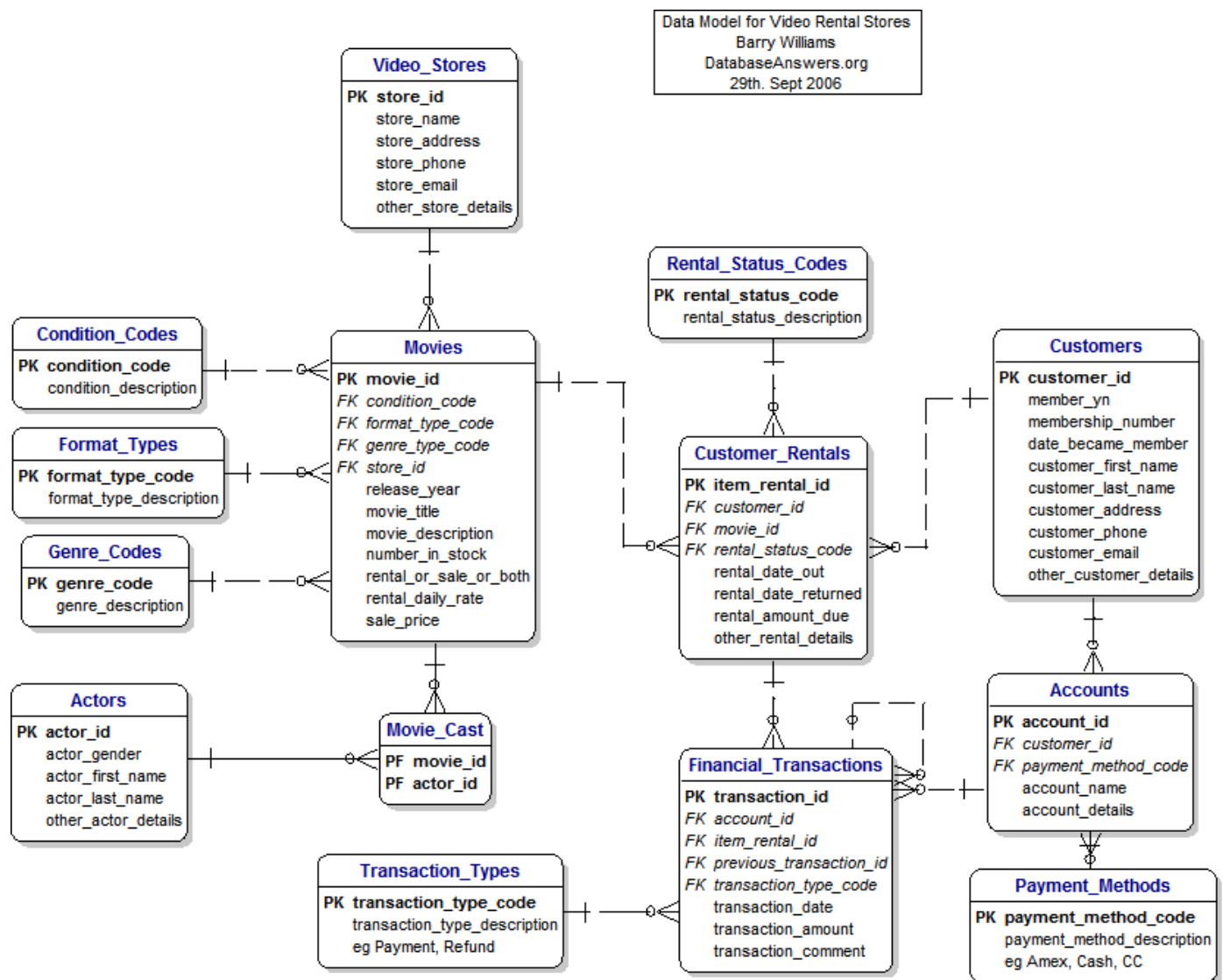


16.2.4 Customers and Invoices

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



16.2.5 Video Rental Stores



16.3 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.

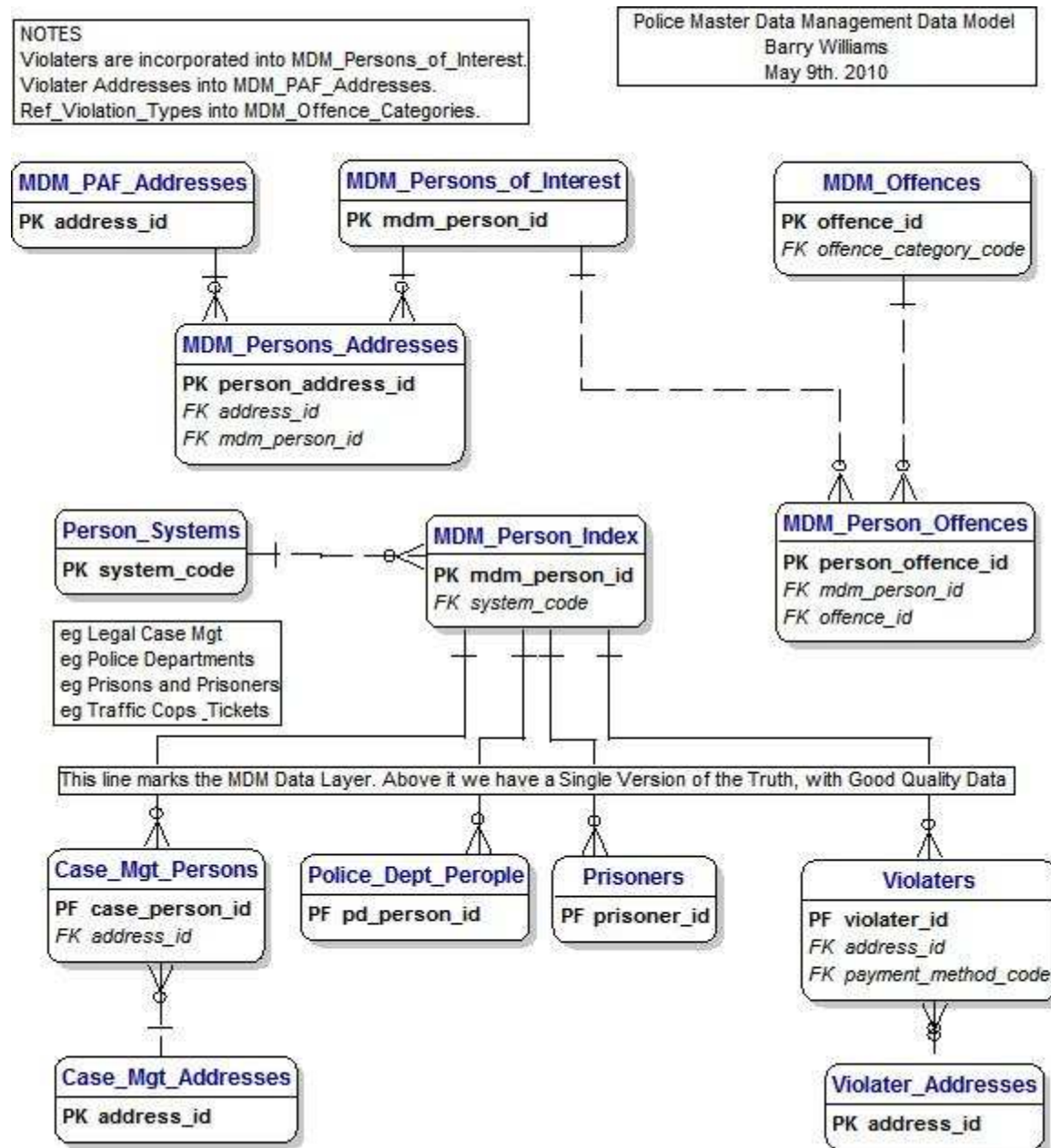
Please email me and let me know at barryw@databaseanswers.org.

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](#).

16.3.1 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.



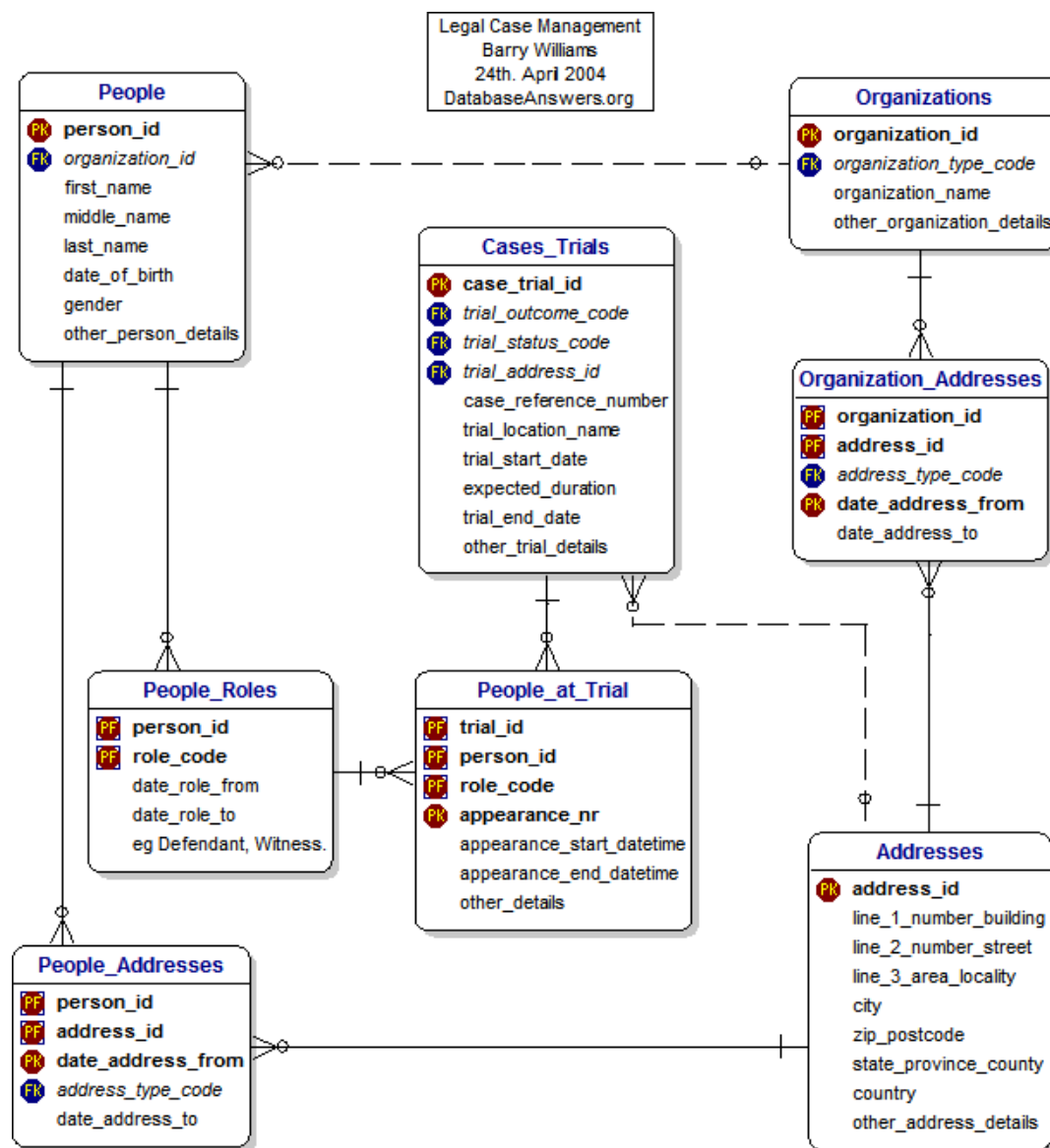
16.3.2 Reference Data

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.

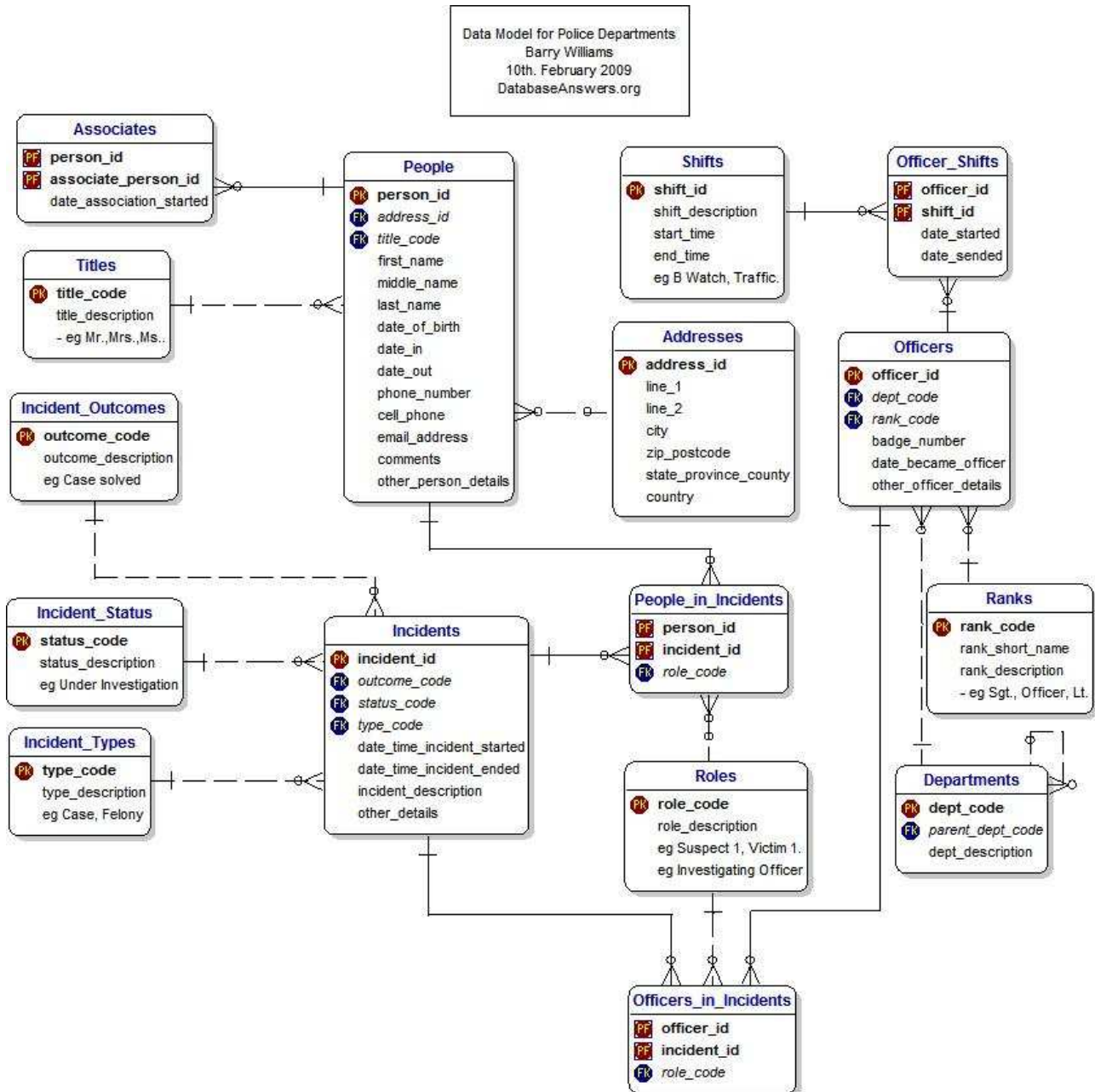
16.3.3 Case Management

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



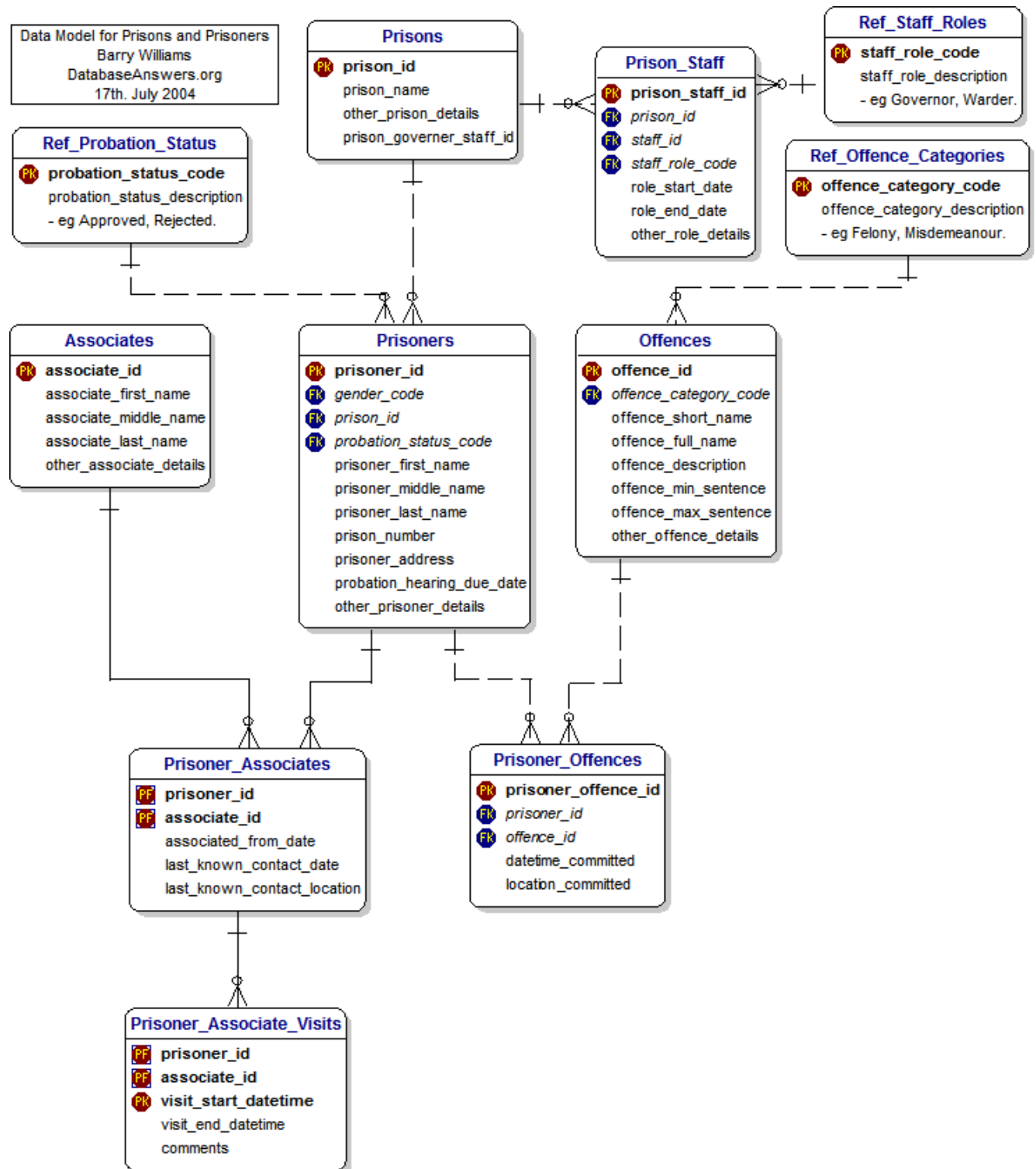
16.3.4 Police Departments

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



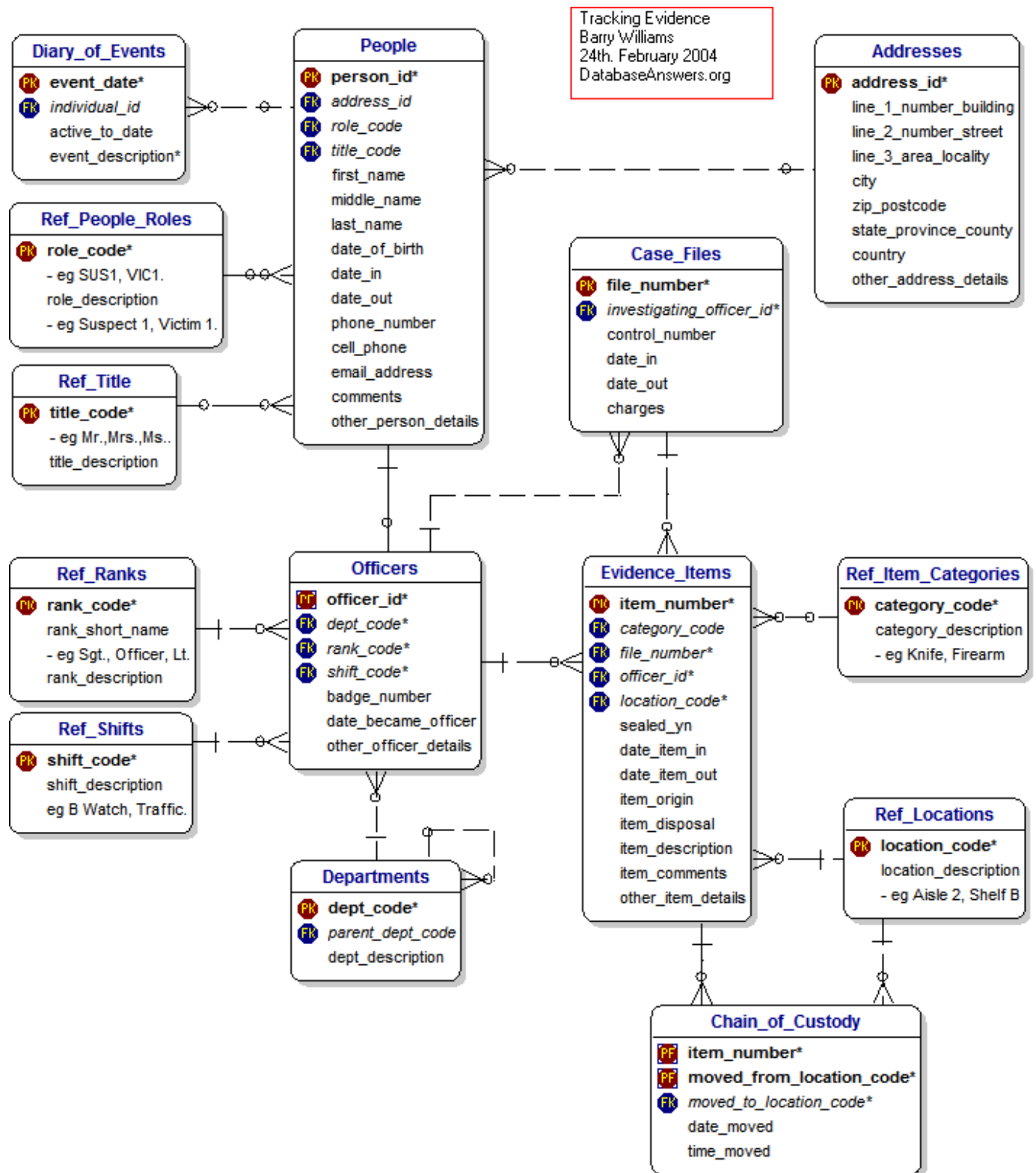
16.3.5 Prisons and Prisoners

This model shows that prisoners commit offences and have associates.



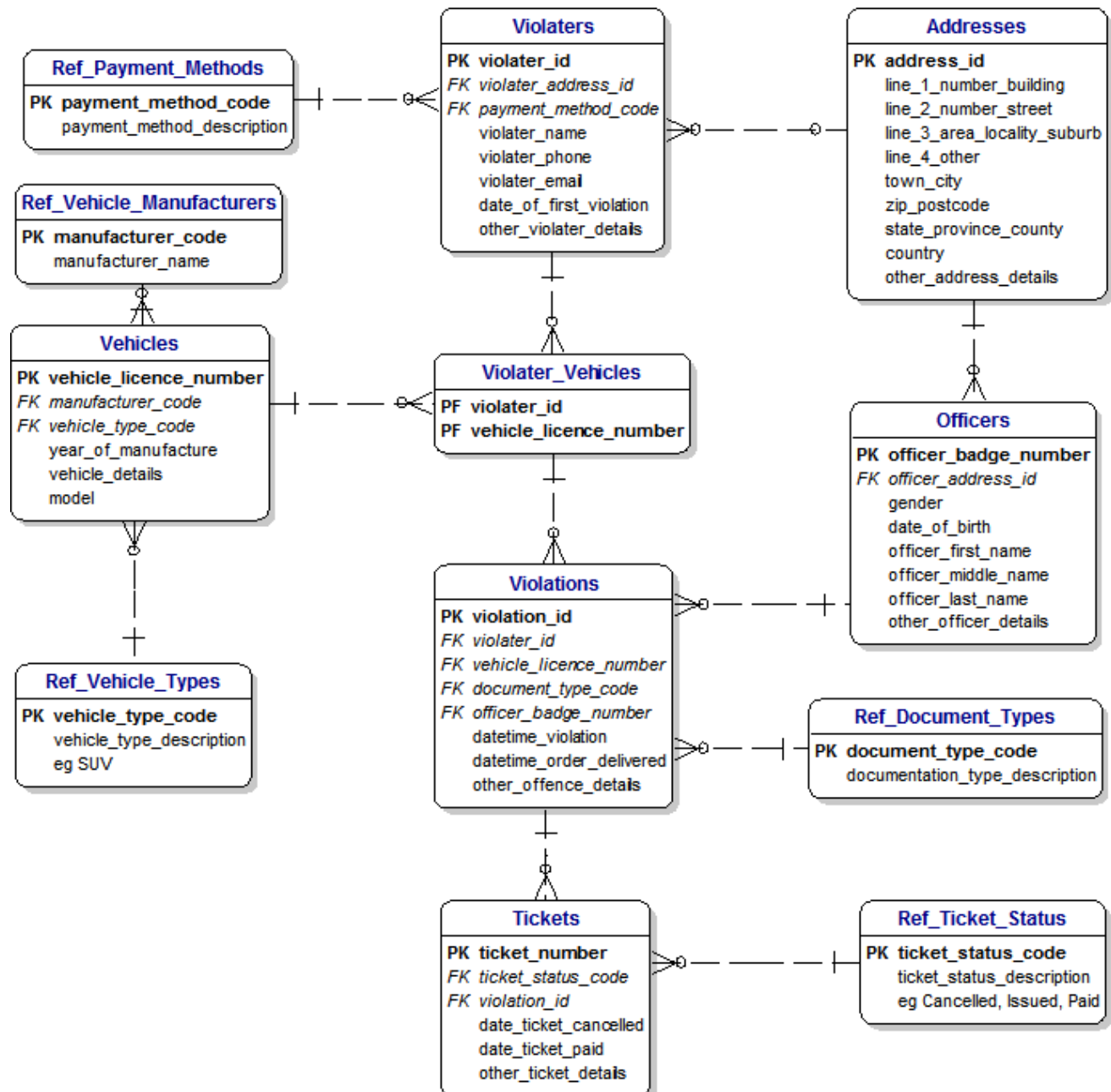
16.3.6 Tracking Evidence

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



16.3.7 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.



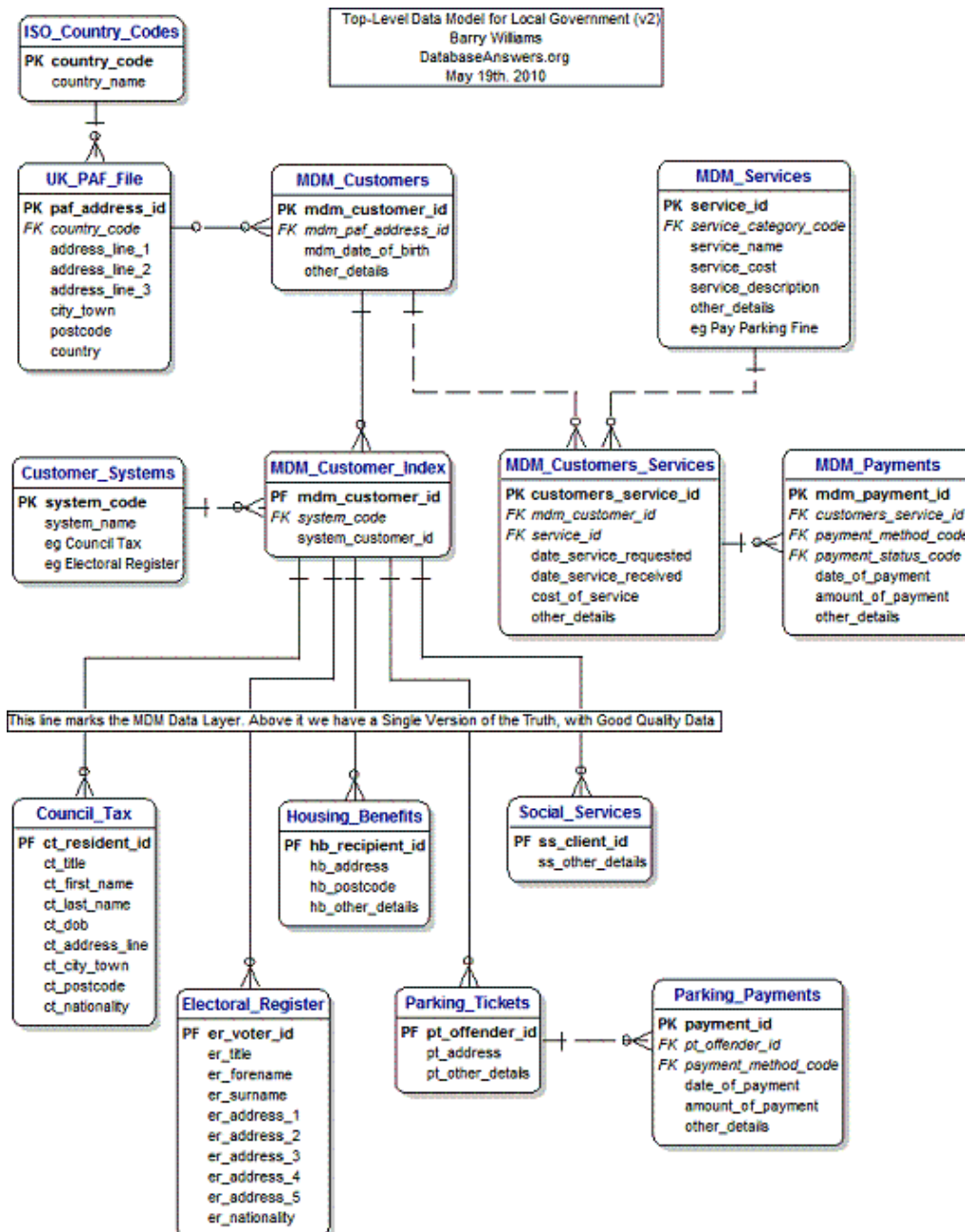
16.4 MDM for Local Government

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

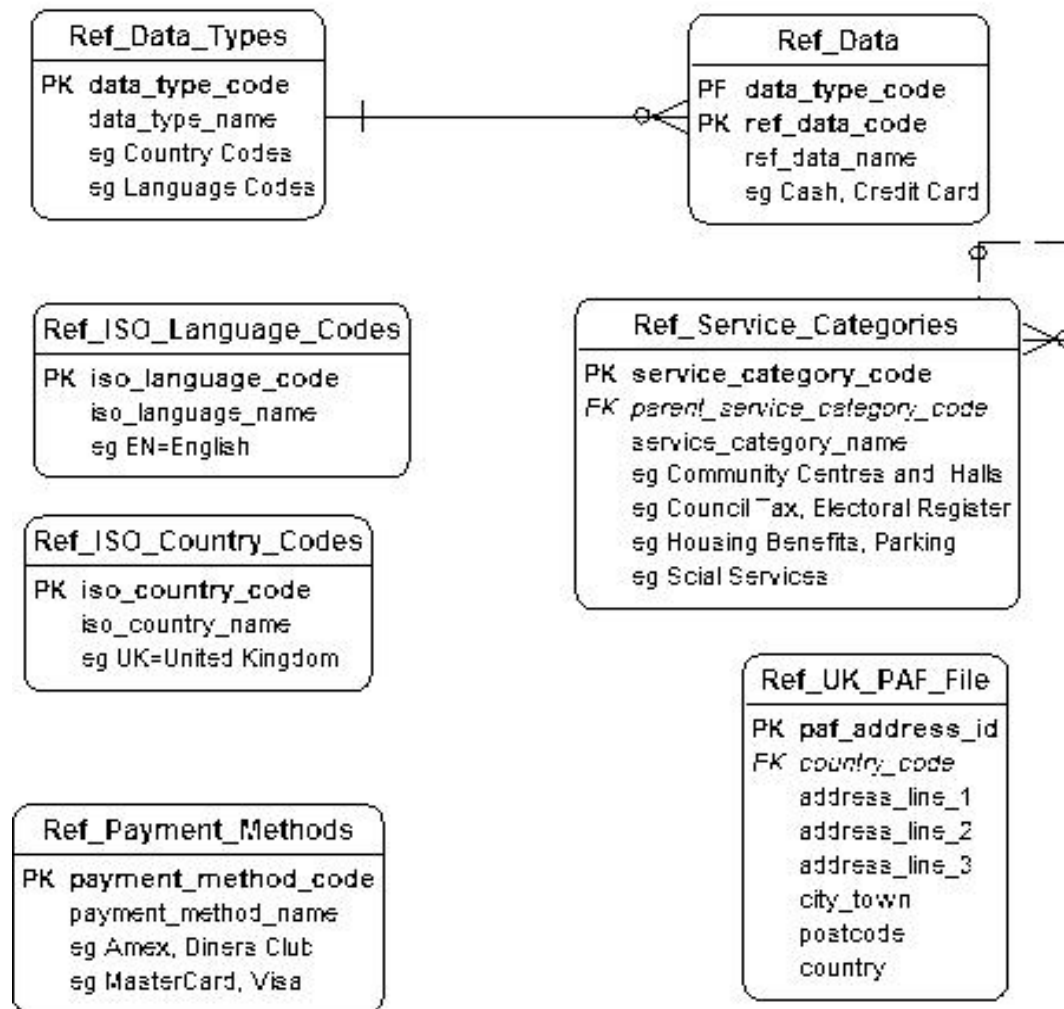
16.4.1 Top-Level Data Model

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.



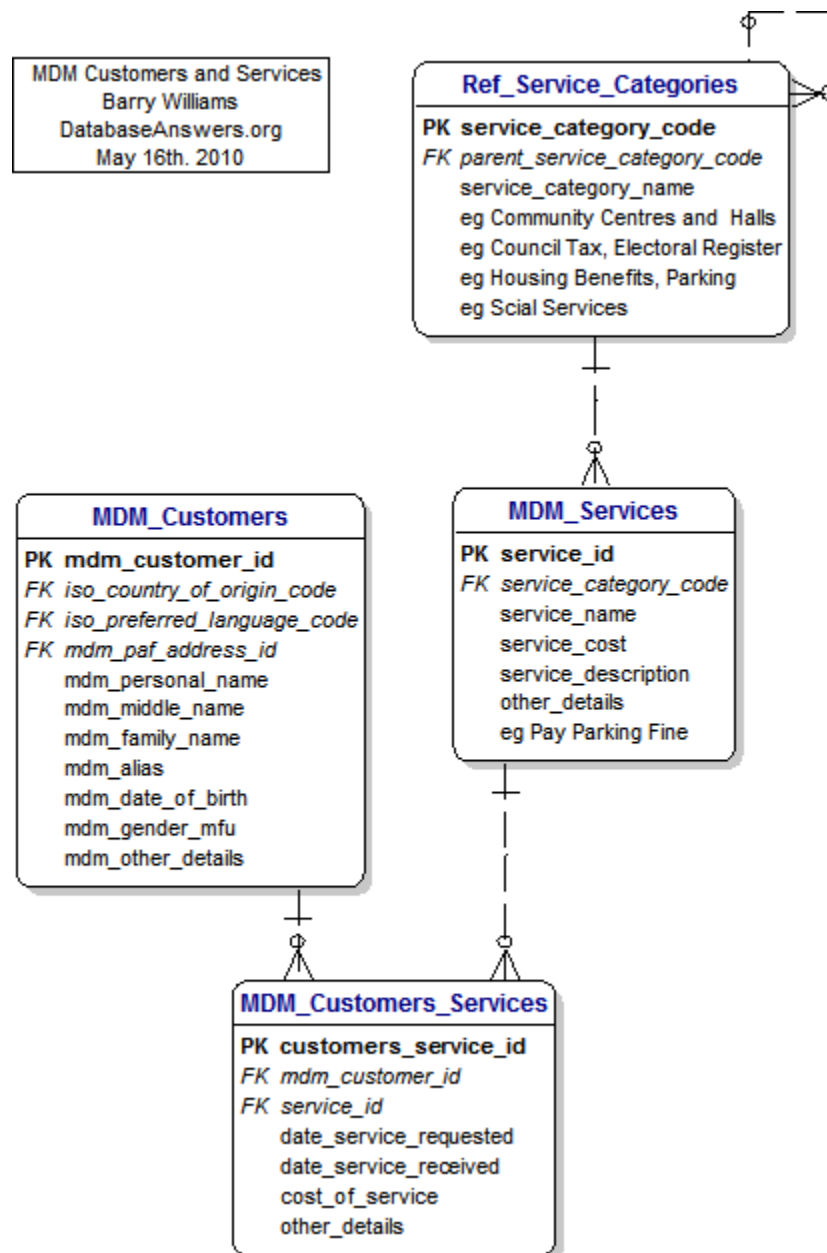
16.4.2 Reference Data

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.



16.4.3 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.



16.4.4 Council Tax

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

Data Model for Council Tax
Barry Williams
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

16.4.5 Housing Benefits

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

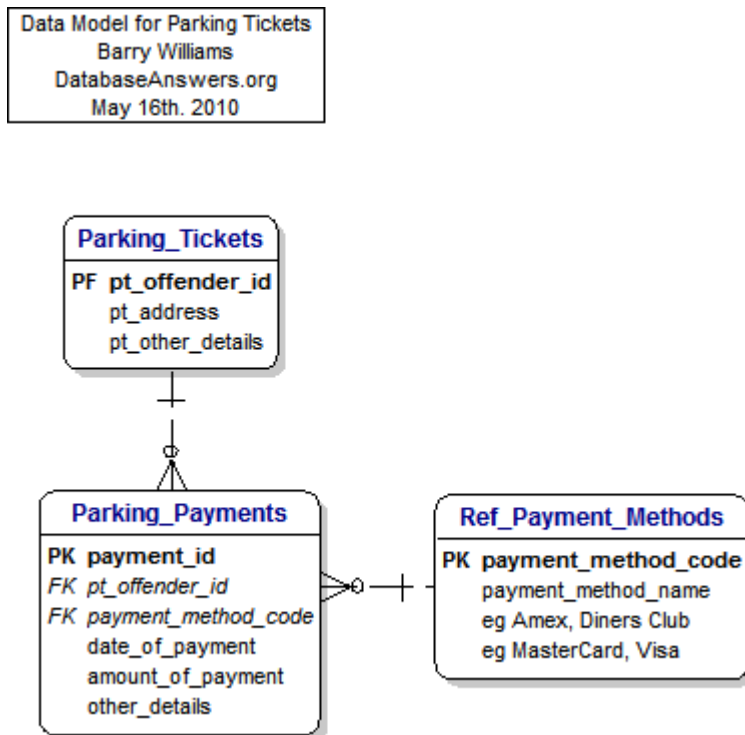
Data Model for Housing Benefits
Barry Williams
DatabaseAnswers.org
May 16th. 2010

Housing_Benefits	
PF	hb_recipient_id
	hb_address
	hb_postcode
	hb_name
	hb_alias
	hb_other_details

16.4.6 Parking Tickets

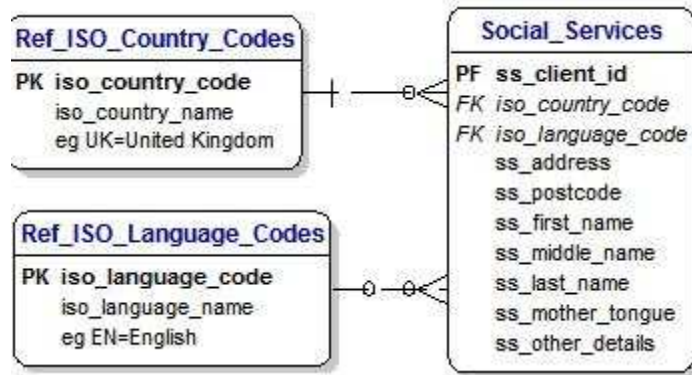
This shows the fields in the tables that are candidates for mapping to the MDM tables.

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.



16.4.7 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.



16.5 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.

17. Build an Enterprise Data Model

17.1 Introduction

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

17.1.1 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

17.1.2 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.

17.1.3 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 catalog to record these models.

Define the naming standards for entities and attributes.

Define the scope of your new model from the user requirements.

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.

17.2 Opening a Pool Hall

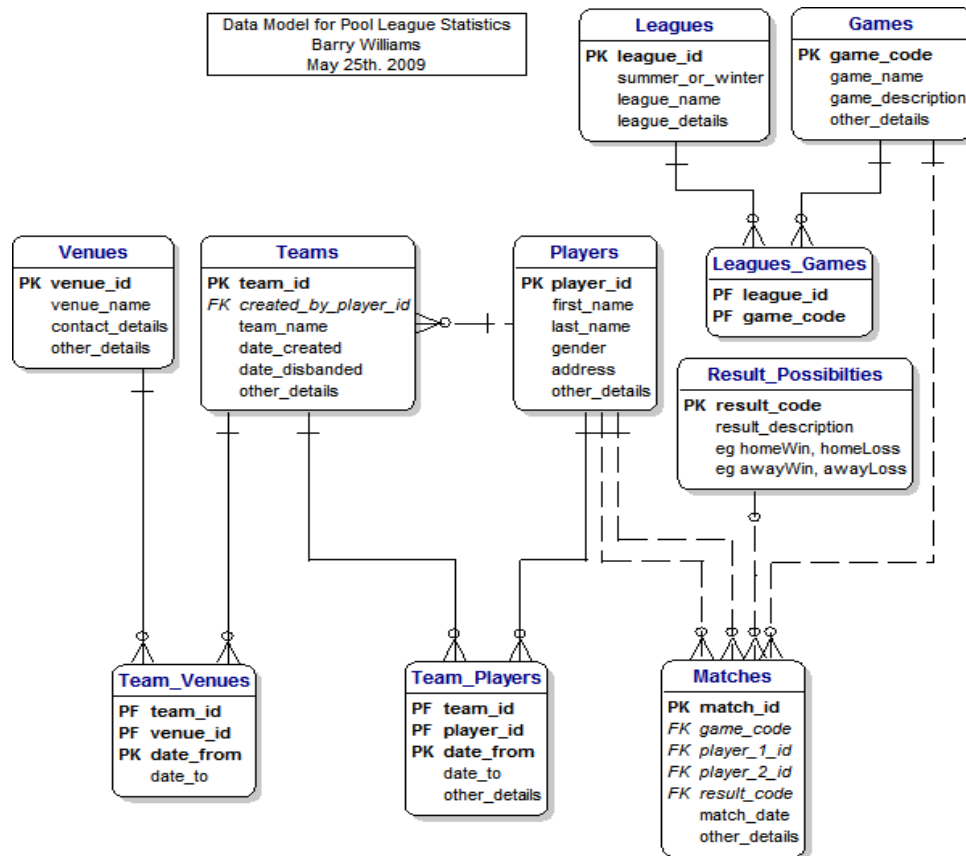
Where do we start with our data model?

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



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

[Reservations:](#)

[Apartment Rentals](#)

[Airline Reservations](#)

[Car Hire](#)

[Cinema Bookings](#)

[Coach Trips](#)

[Doctors Practice](#)

[Driving Schools](#)

[Event Reservations](#)

[Flights for Children's Charities](#)

[Hairdressers Appointments](#)

[Health Centers](#)

[Holiday Homes](#)

[Hotel Reservations](#)

[Limousine Services](#)

[Online Scheduling](#)

[Railway Reservations](#)

[Restaurant Bookings](#)

[Sports Centers](#)

[Ticketmaster](#)

[Vehicle Rental](#)

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:

[Apartment Rentals](#)

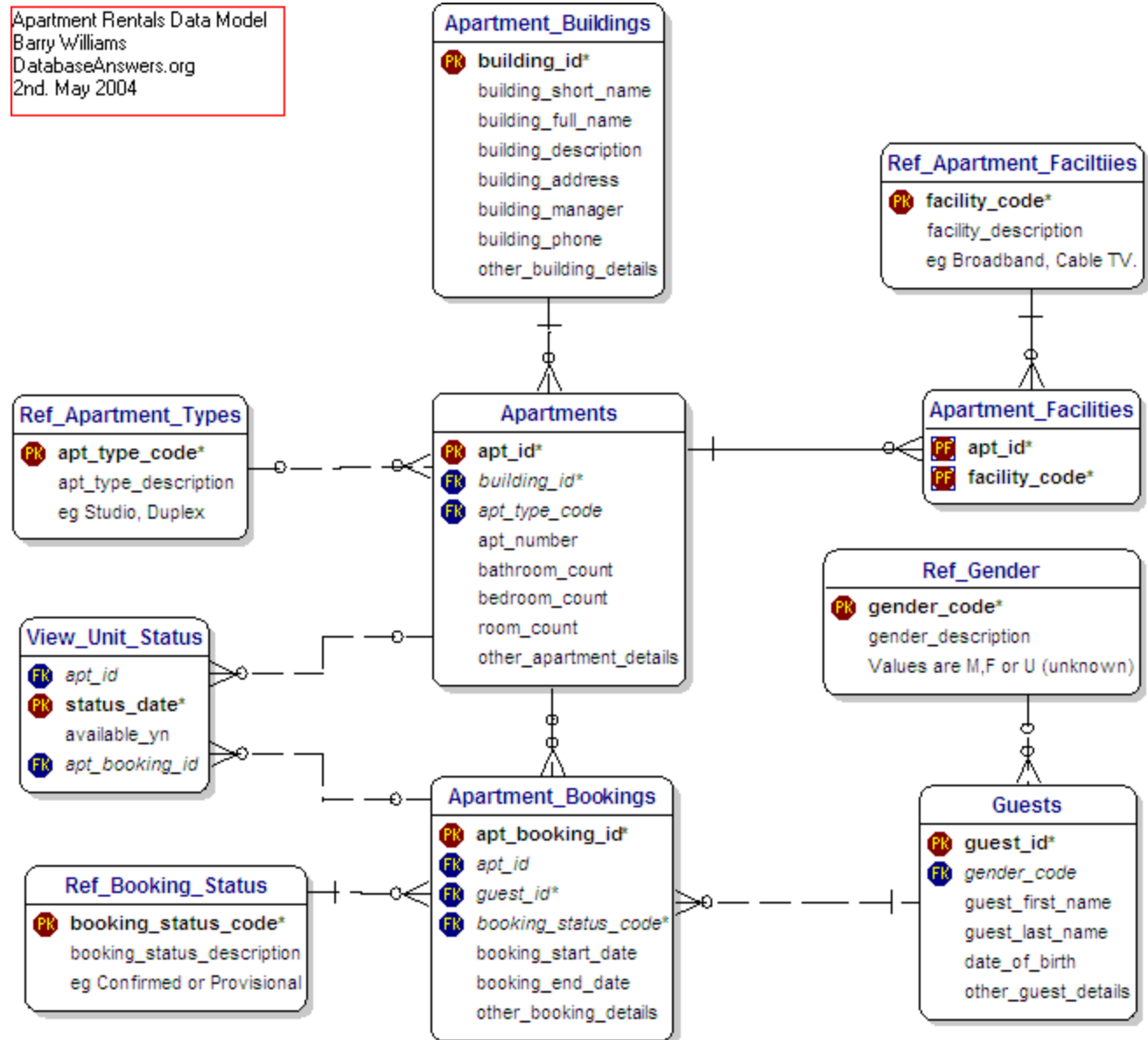
[Driving Schools](#)

[Hairdressers Appointments](#)

[Railway Reservations](#)

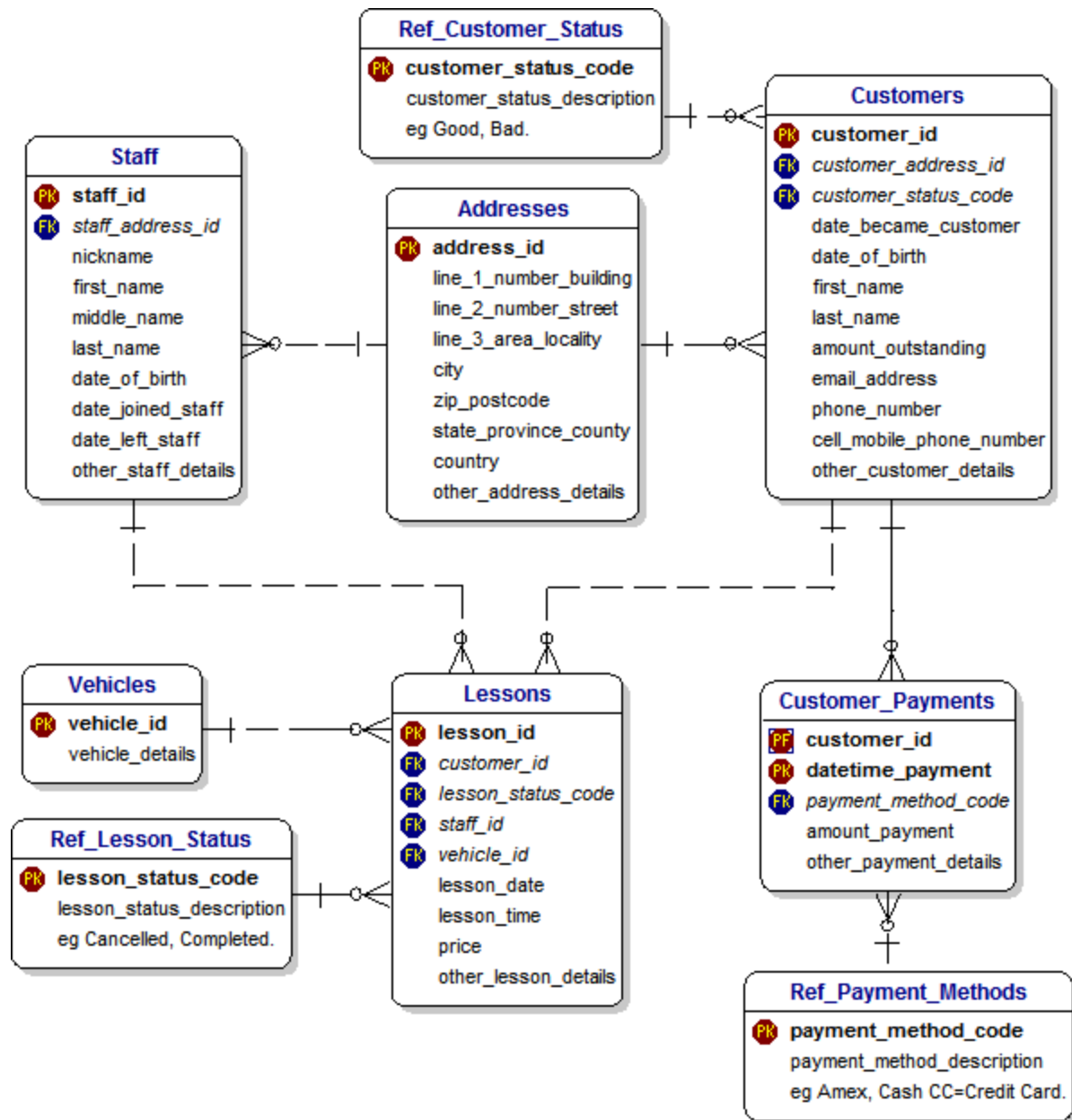
[Sports Centers](#)

1. Apartment Rentals

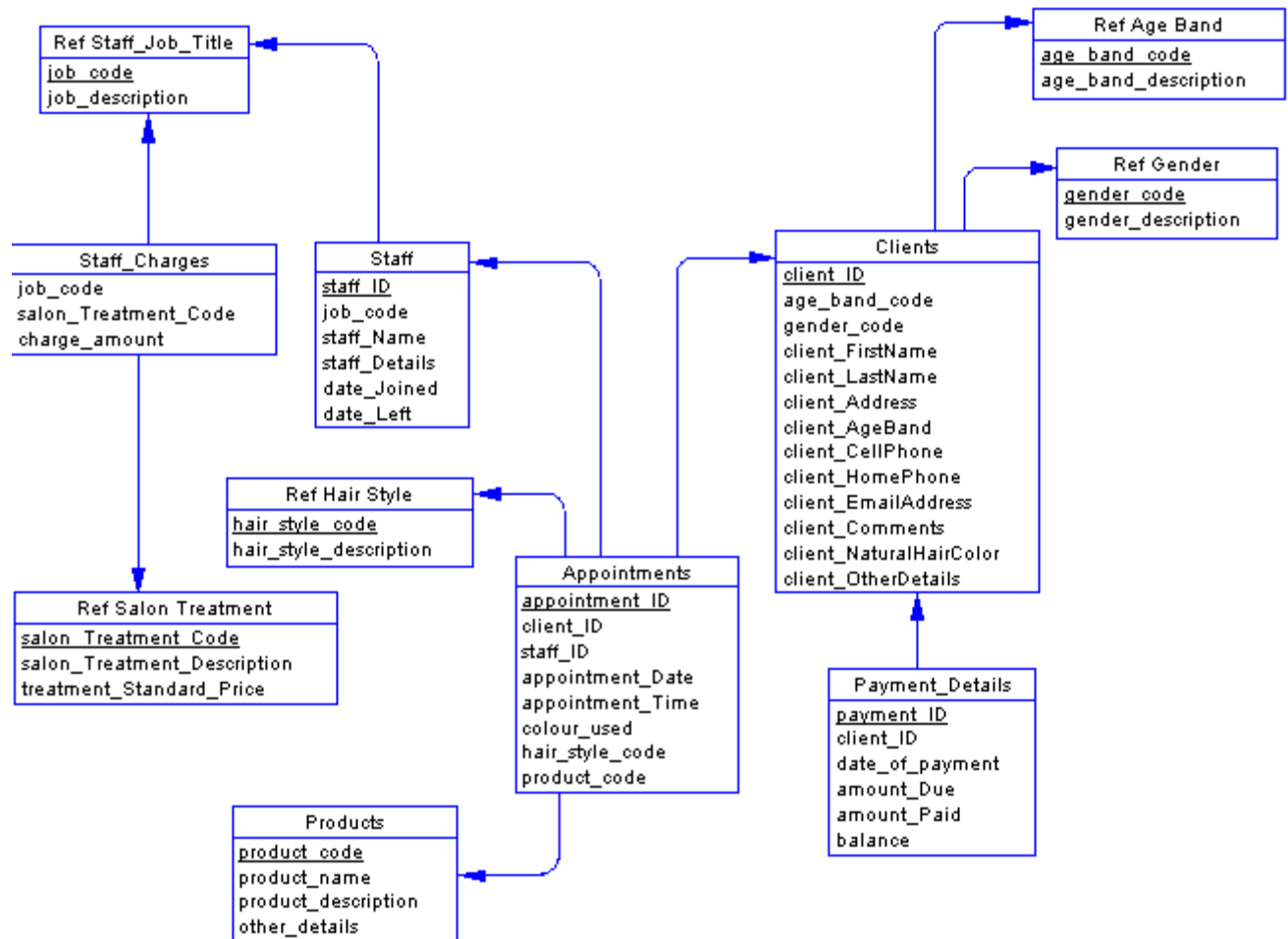


2. Driving Schools

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

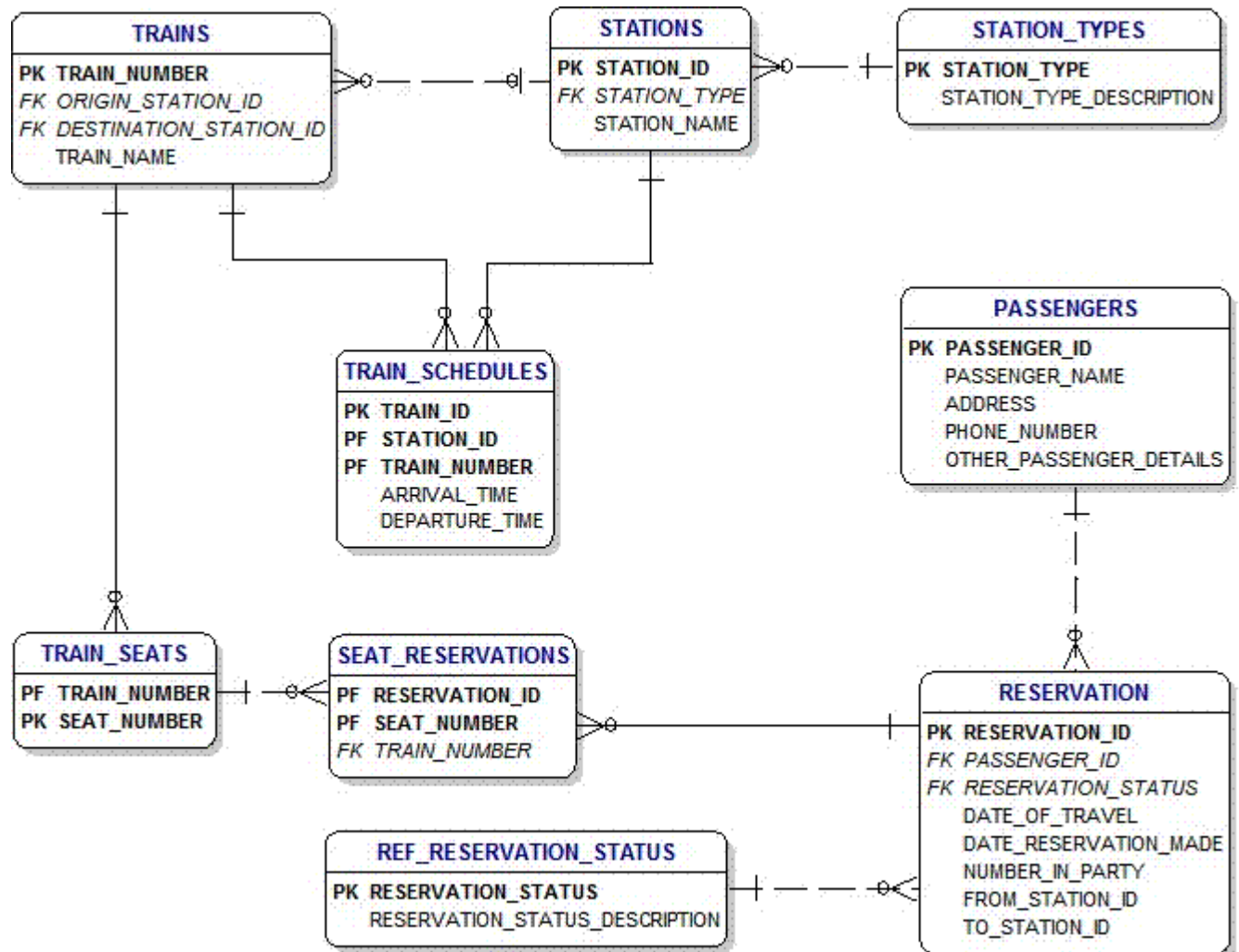


3. Hairdressers Appointments



4. Railway Reservations

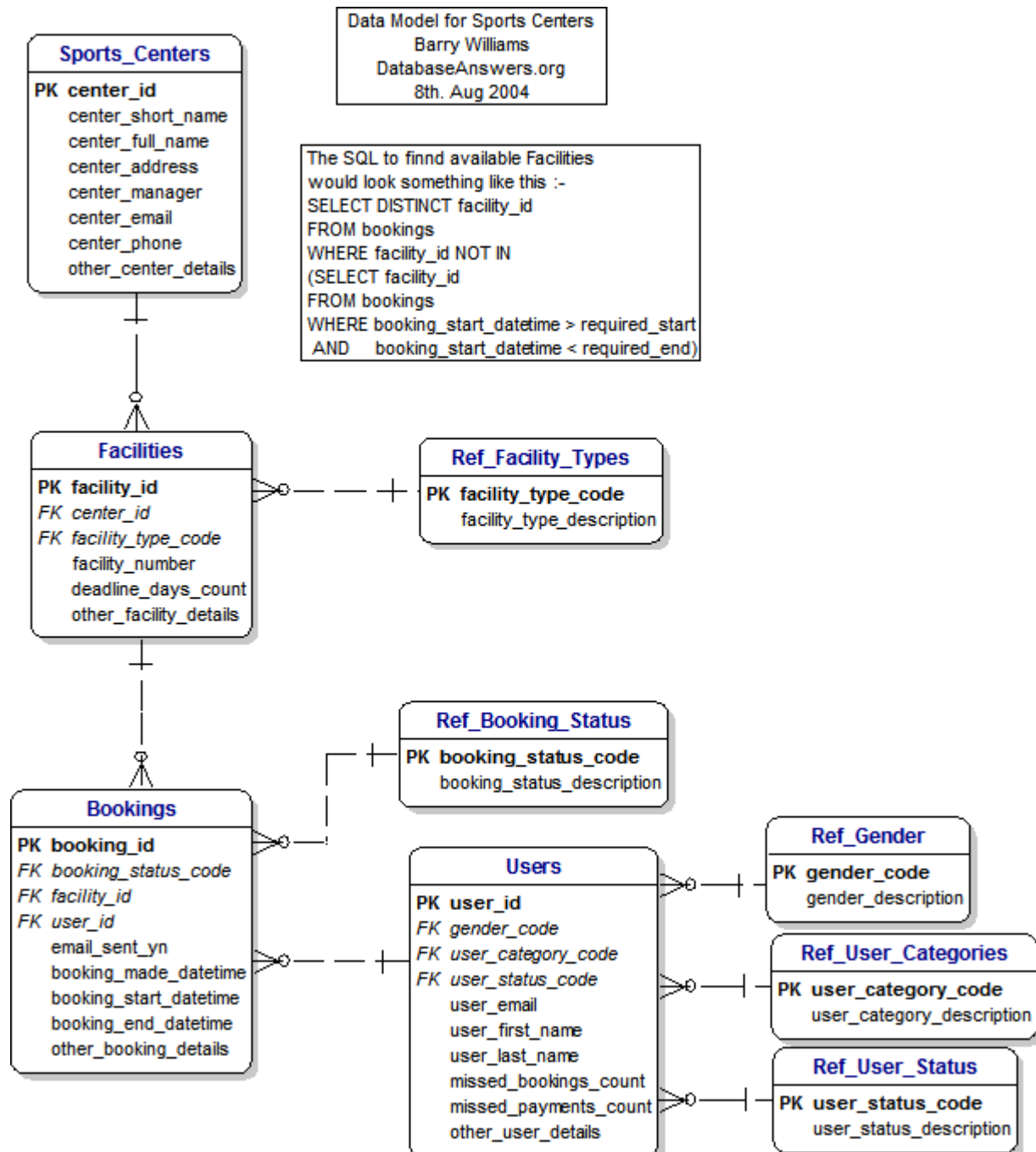
This looks too complicated and includes things that we don't need, such as stations.



5. 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.



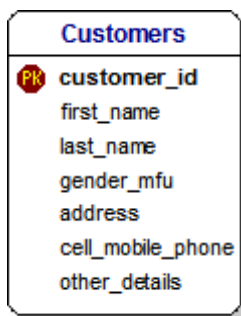
Step 1. Start with the Customer

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:

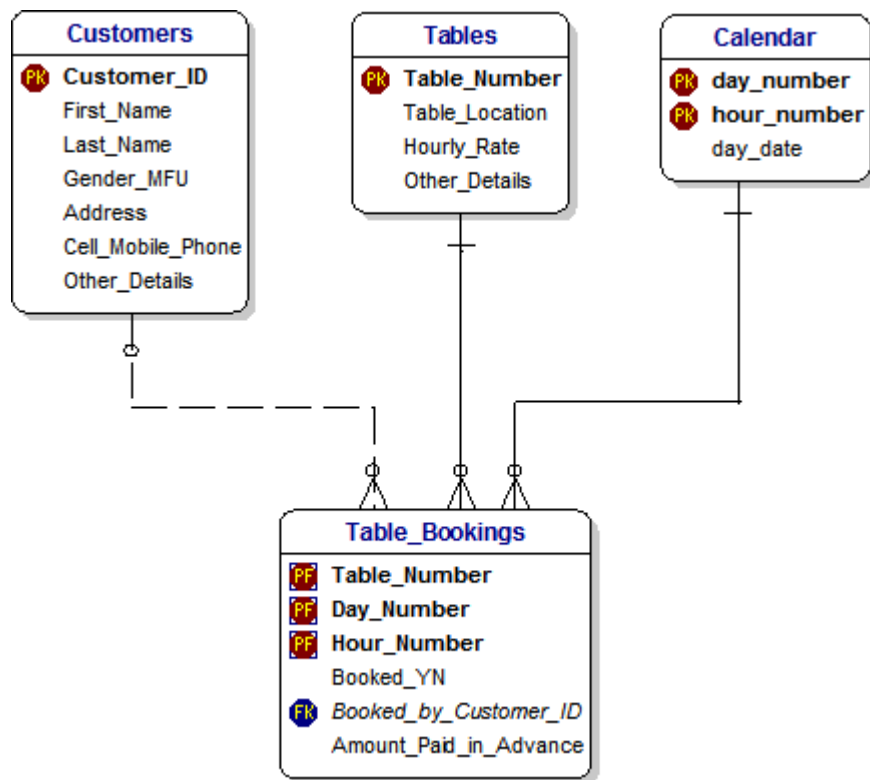
- http://www.databaseanswers.org/data_models/customers_and_credit_cards/customers_and_credit_cards_with_attributes.htm

So our data model consists of just this customer entity:



Step 2. Add Tables and Bookings

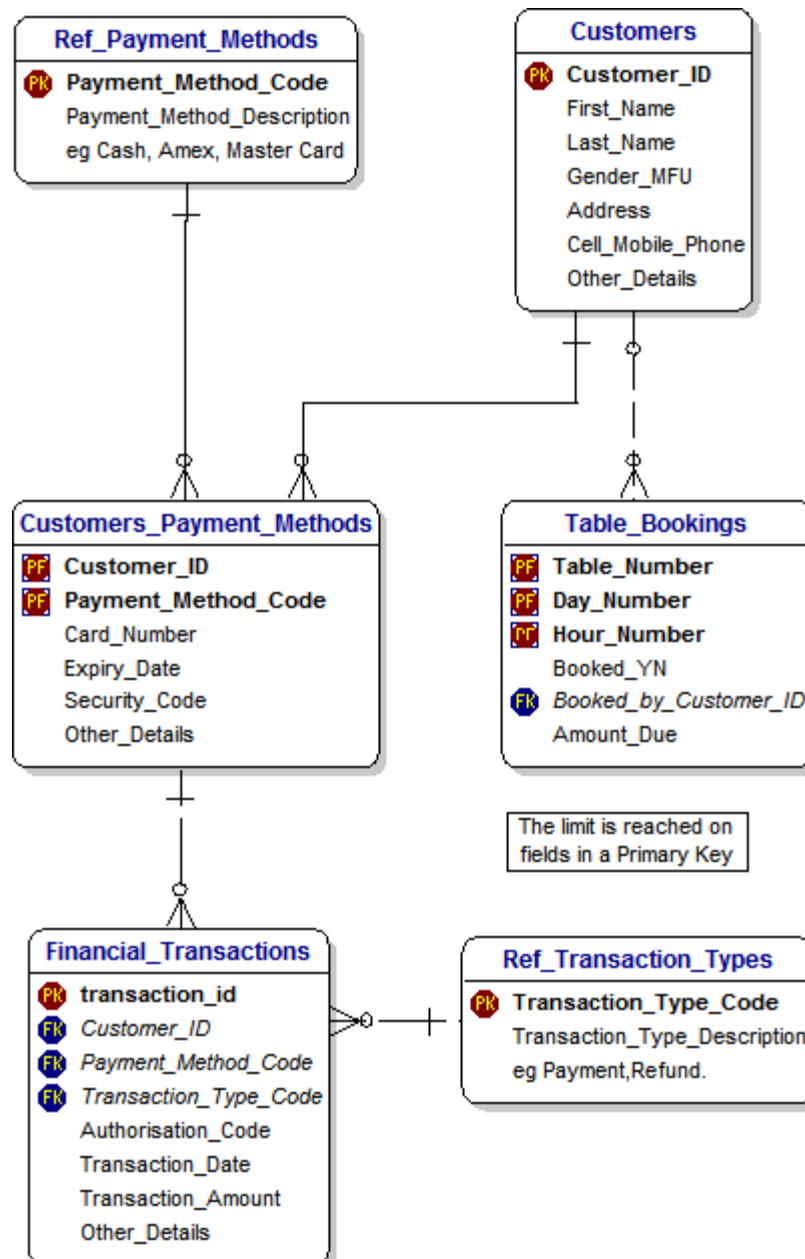
This is done in a style consistent with the Sports Centers model. Full payment is made in advance.



Step 3. Add Customers and Payment Methods

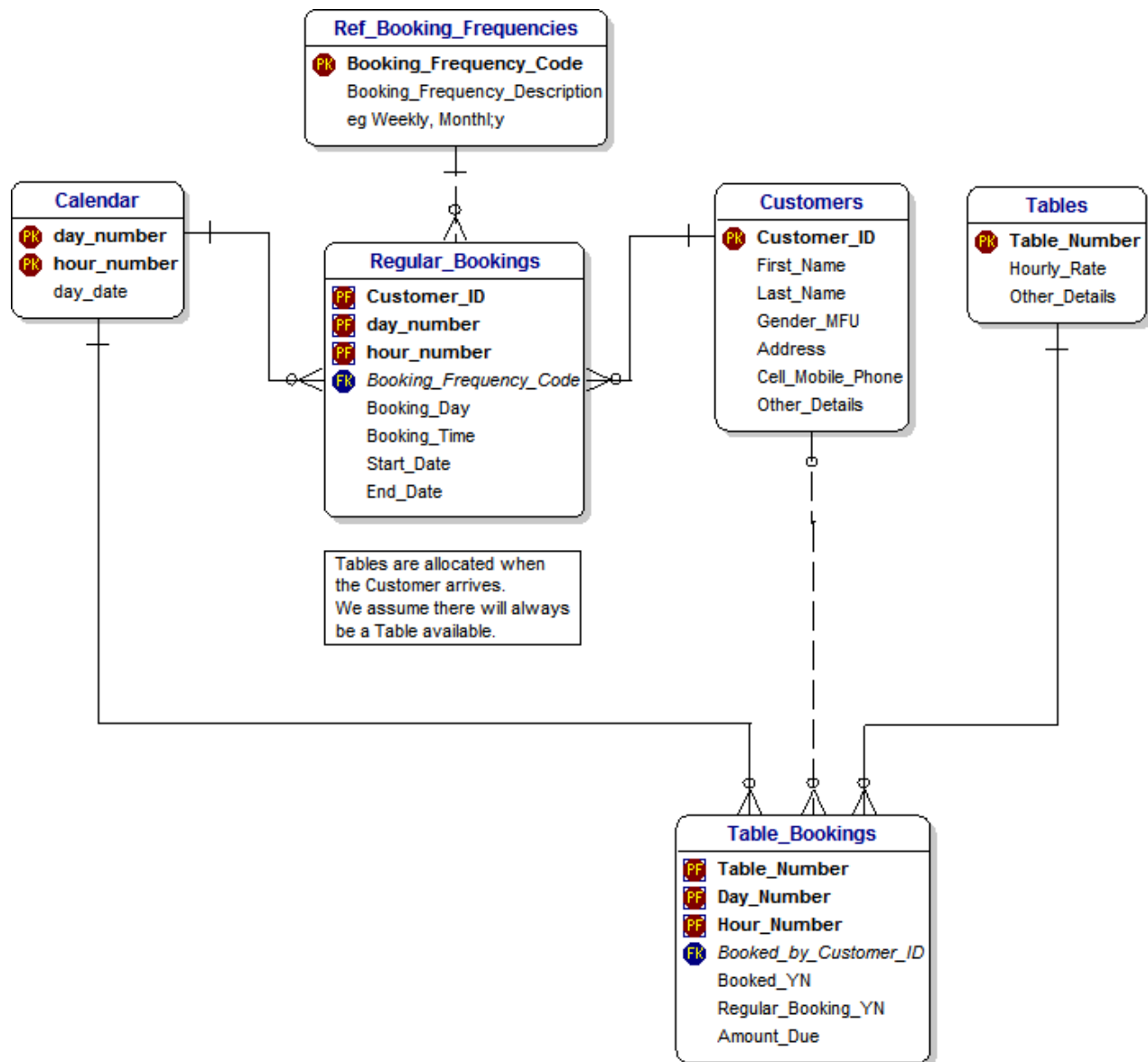
Then we add payments methods from the customers and payments model:

- http://www.databaseanswers.org/data_models/customers_and_payments_e_govt/customers_and_payments_subject_area.htm



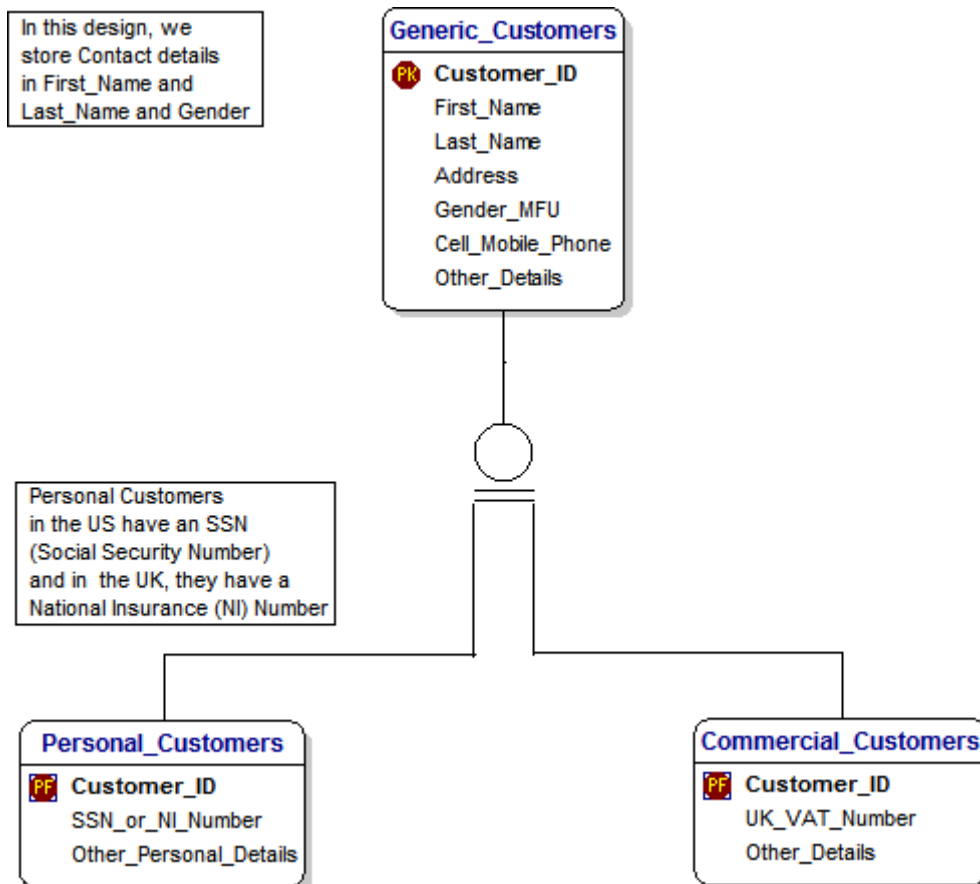
Step 4. Add Regular Bookings

This step adds repeat bookings. Booking Frequencies table is at the top because logically that is where it belongs. However, it gives it an importance that it does not deserve.



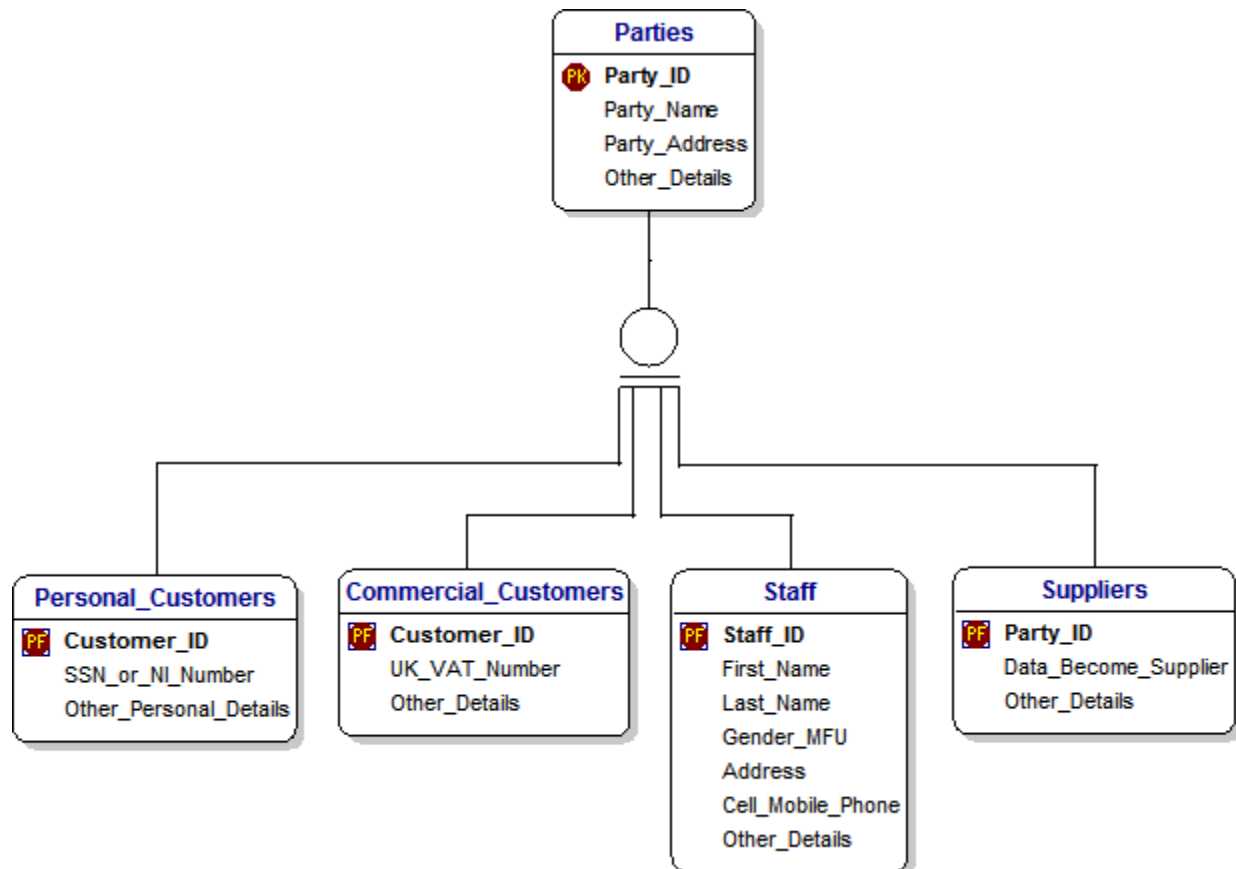
Step 5. Add Inheritance

Generic customers contain common Attributes from personal and commercial customers.



Step 6. Let's have a Party

This step introduces inheritance and the concept of parties, which are a *super-type*. Personal and commercial customers are *sub-types*, along with staff and suppliers.



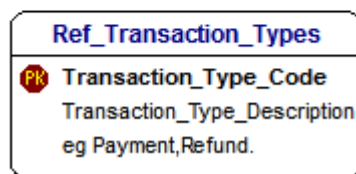
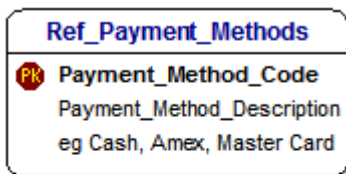
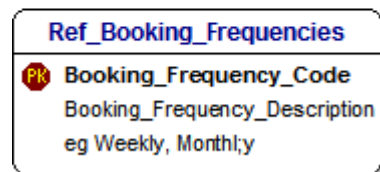
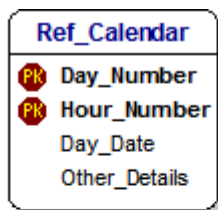
Step 7. Reference Data

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

It is good practice to put 'Ref_' at the beginning of every entity or table. This makes it very easy to identify the reference data.

Some data might be considered reference data under certain circumstances. For example, a calendar is always reference data because it is predictable and never changes.

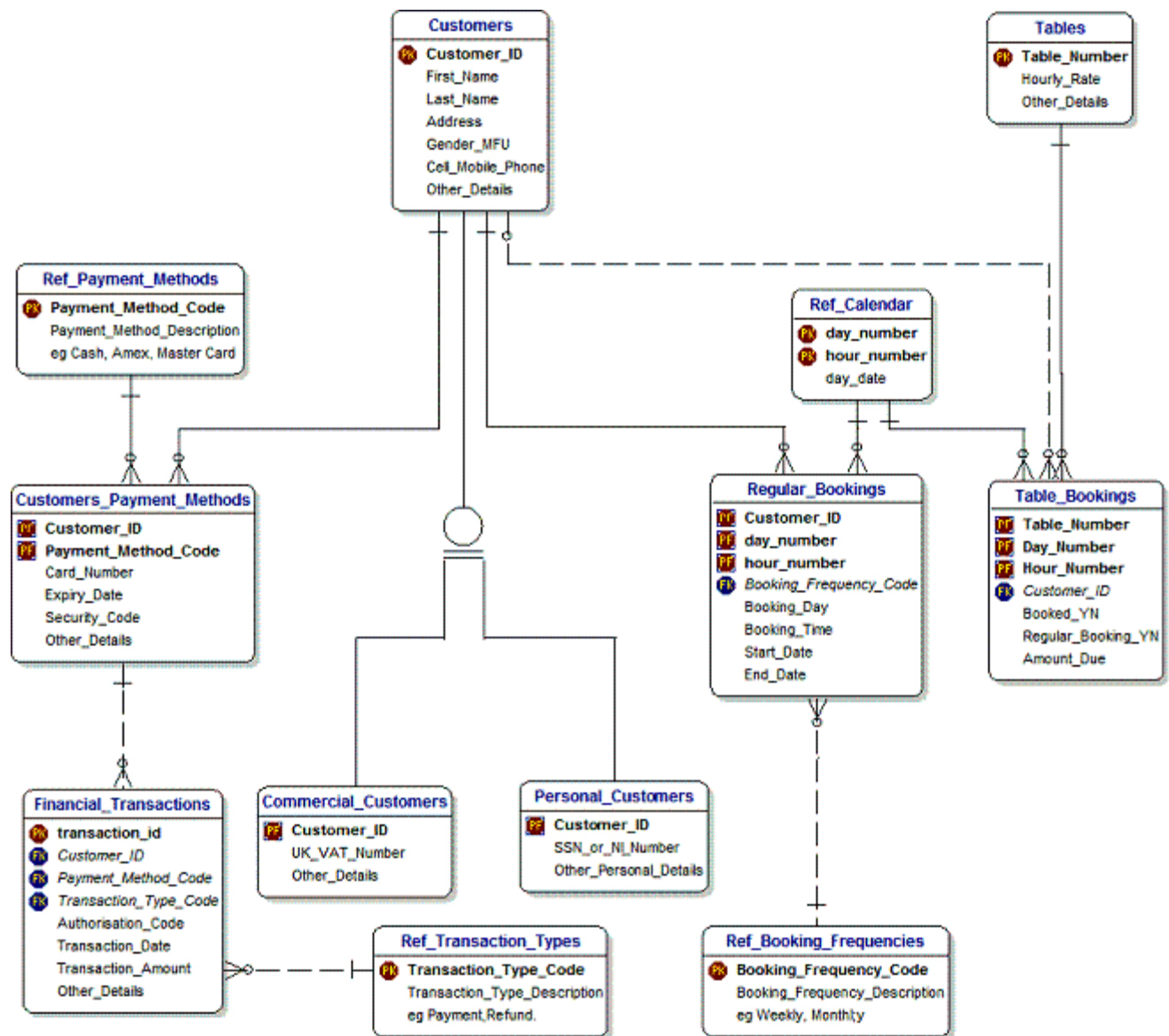
Products, on the other hand, could change regularly and should not be considered reference data. For example, products in finance or banking could change on a weekly basis and are therefore not reference data.



Step 8. The Complete Model

This diagram shows all the tables, all the keys and attributes, including reference data.

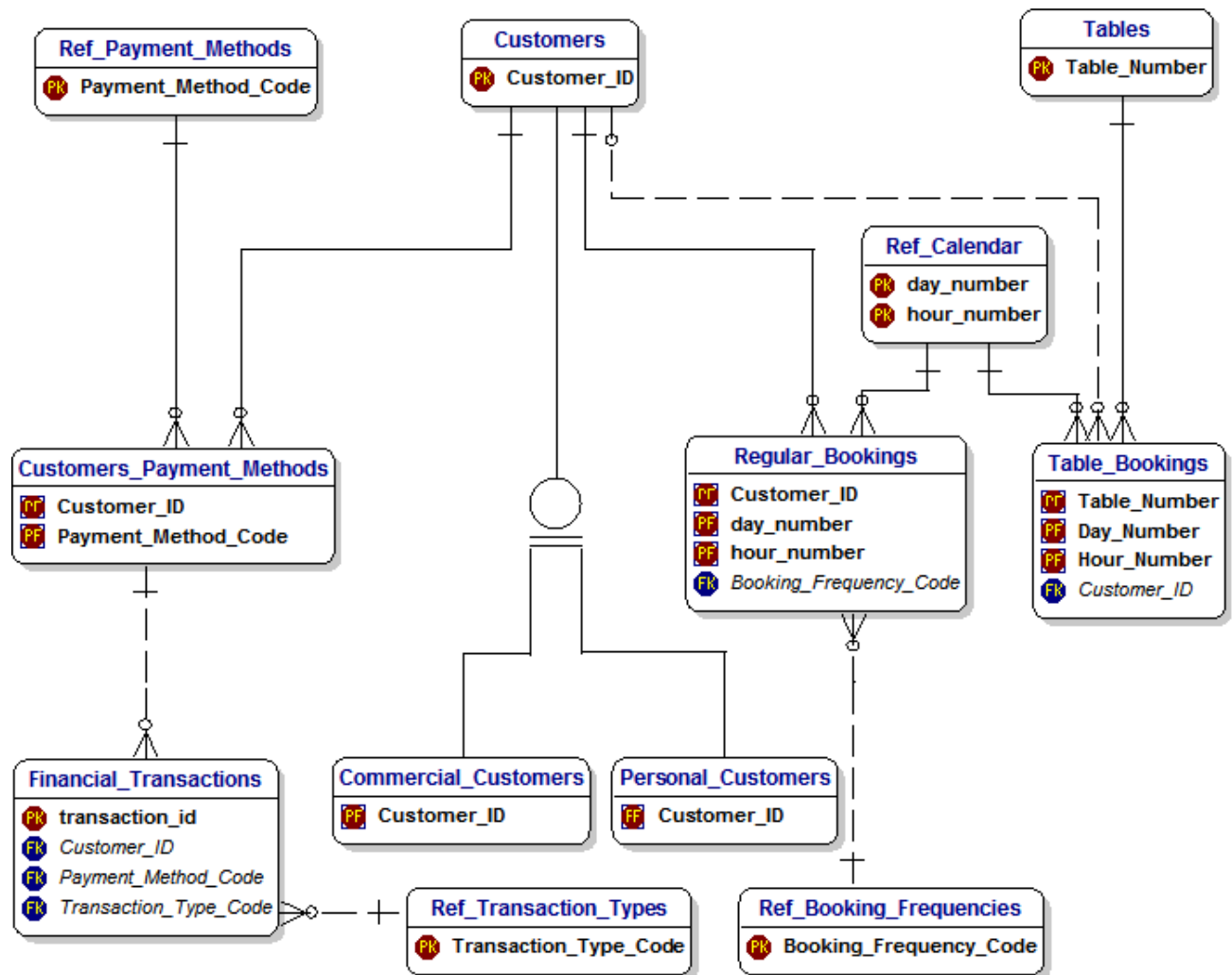
It is suitable for discussion with developers, data analysts and other data modelers who want to see all the details of every table.



Step 9. The Complete Model - Showing Key Fields Only

This shows all the tables, with just the keys. This is much easier to understand at a glance.

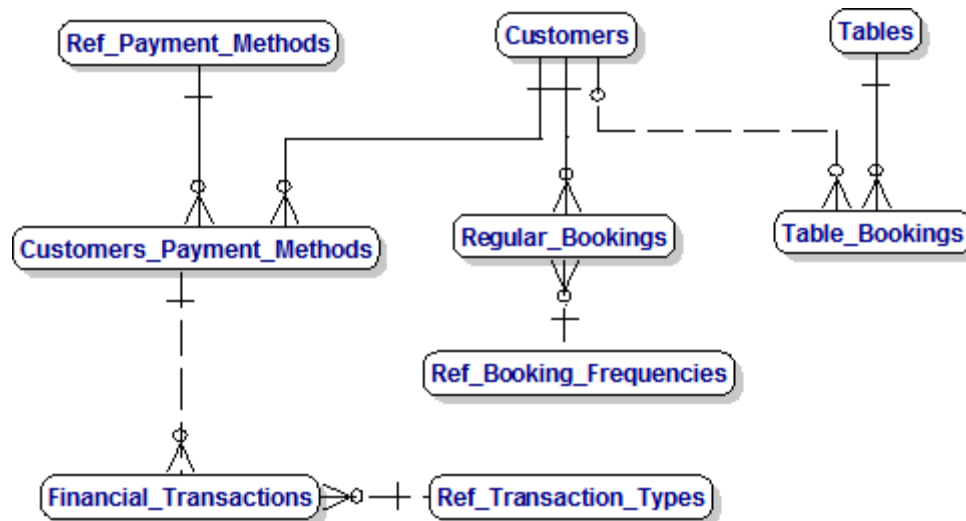
This is suitable for discussion with developers, data analysts and other data modelers.



Step 10. 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

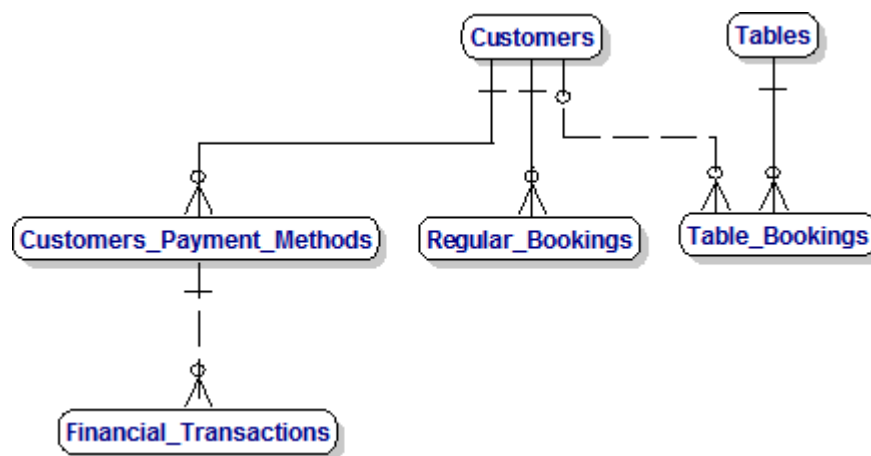
This is even easier and is usually acceptable to senior stakeholders. They will understand the concept of reference data but do not need to see it shown explicitly.

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 12. Top-Level and Subject Area Models

The subject areas include:

- Bookings
- Customers
- Payments
- Data Warehouse

17.2.1 Top-Level Model

The Top-Level Model goes here.

17.2.2 Subject Area Models

The Subject Area Model goes here.

17.2.3 Bookings

The Bookings Model goes here.

17.2.4 Customers

The Customers Model goes here.

17.2.5 Payments

The Payments Model goes here.

17.2.6 Data Warehouse

The Data Warehouse Model goes here.

17.3 Opening a Funeral Home

Here is another finished product:

- http://www.databasanswers.org/data_models/funeral_homes/index.htm

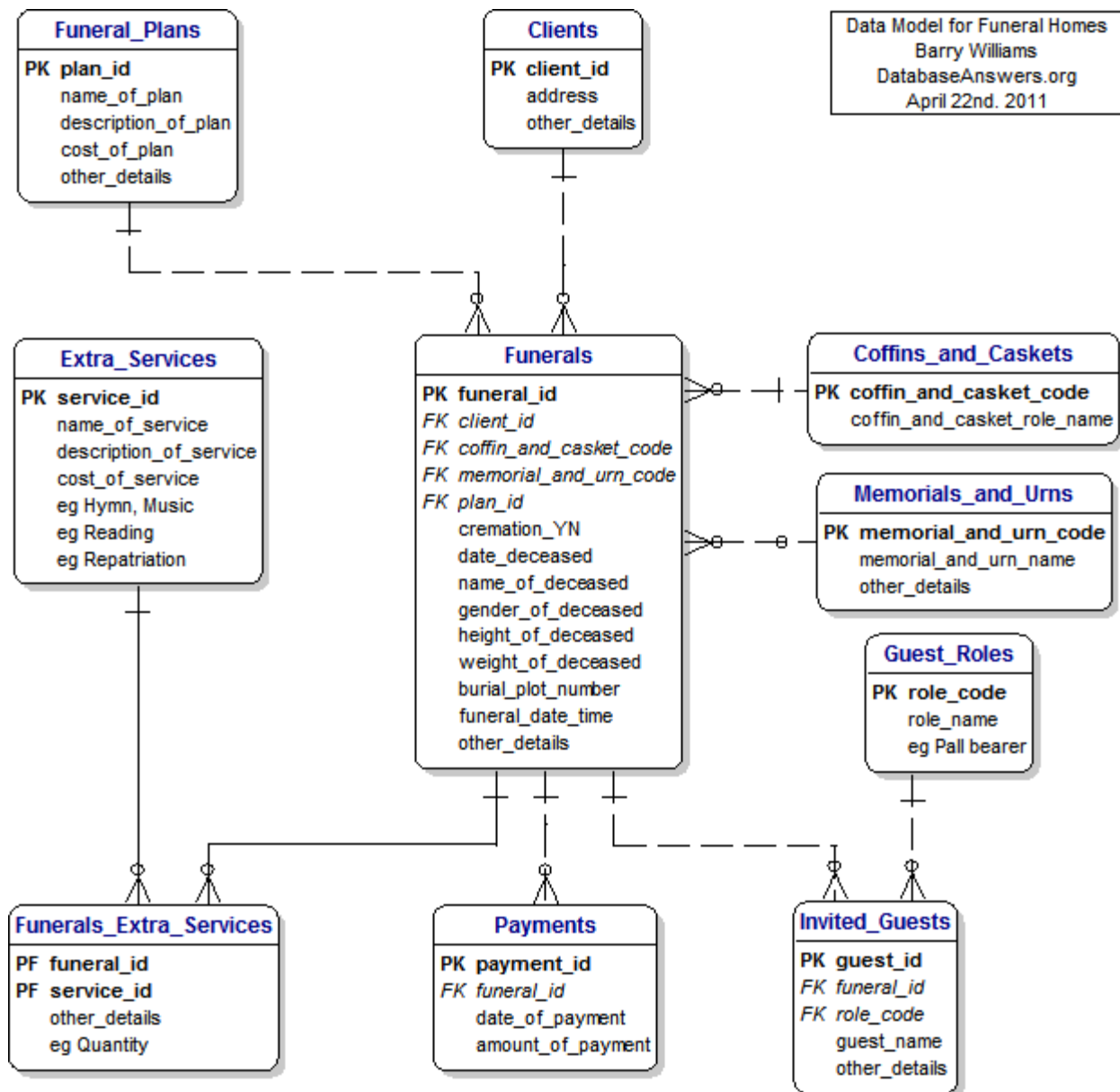
Step 1. Identify the 'Things of Interest'

These include:

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

How are they related?

Clients request *funerals* with optional *services*.



17.4 Making a Movie

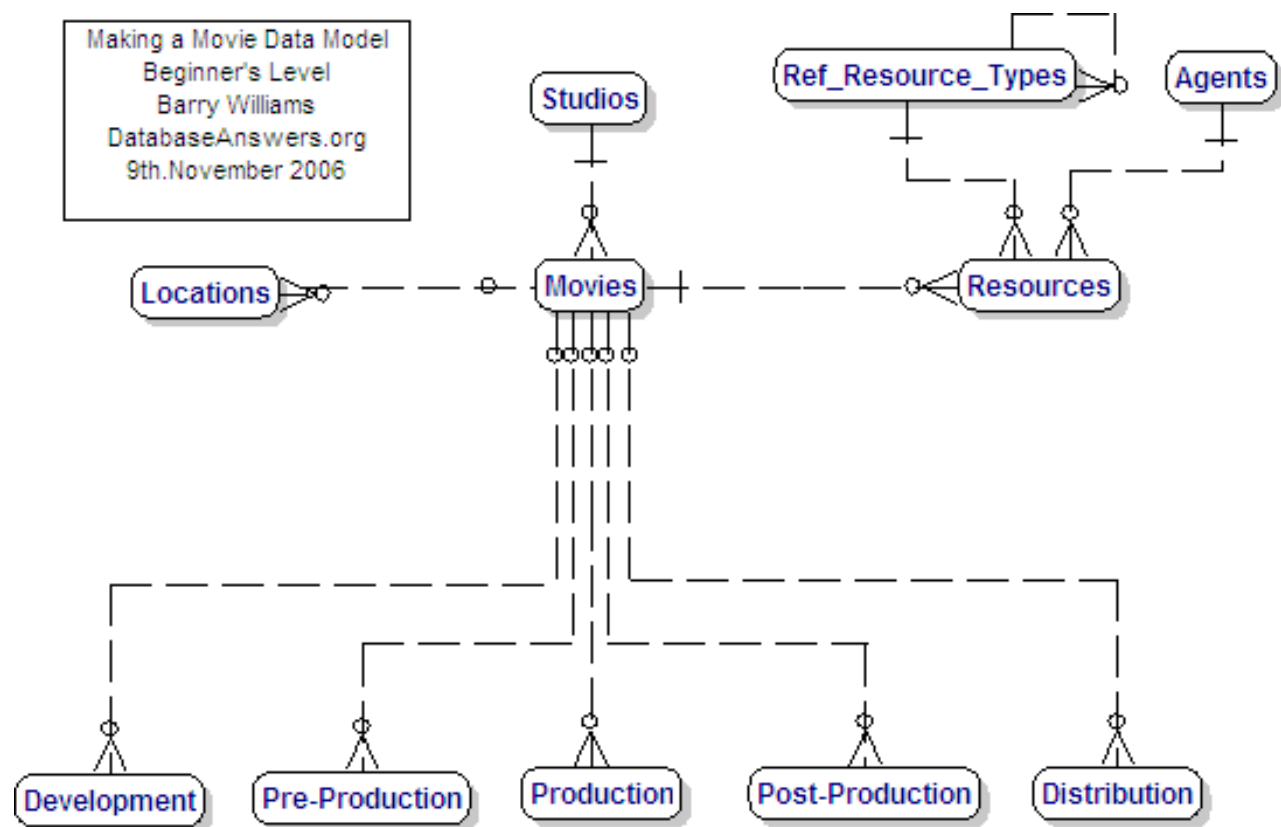
17.4.1 A Simple Model

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.



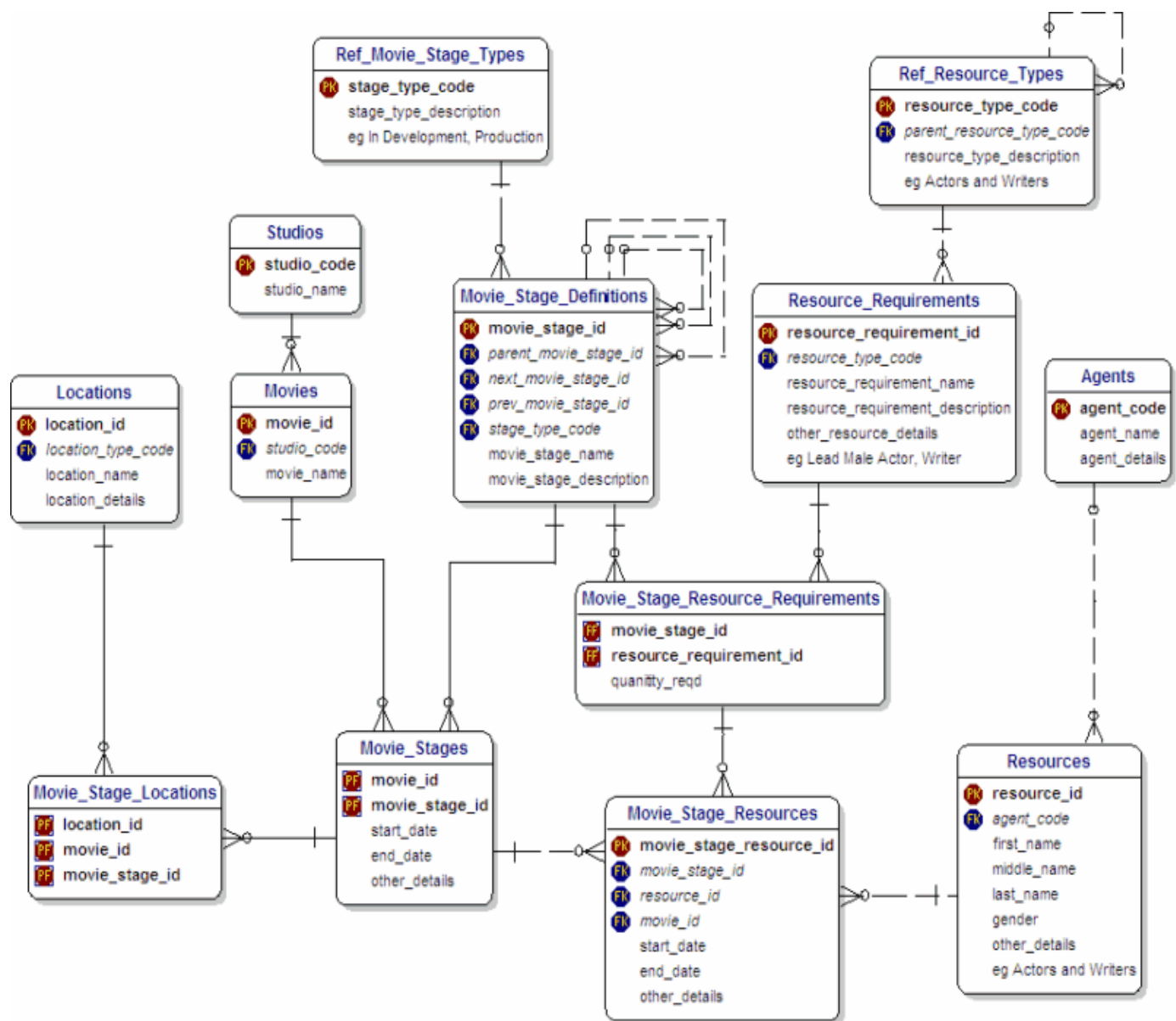
17.4.2 A Complex Model

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.



17.5 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)

18. A Single View of the Truth

18.1 Introduction

18.1.1 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.

18.1.2 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 Catalogs:

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

18.1.3 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'.

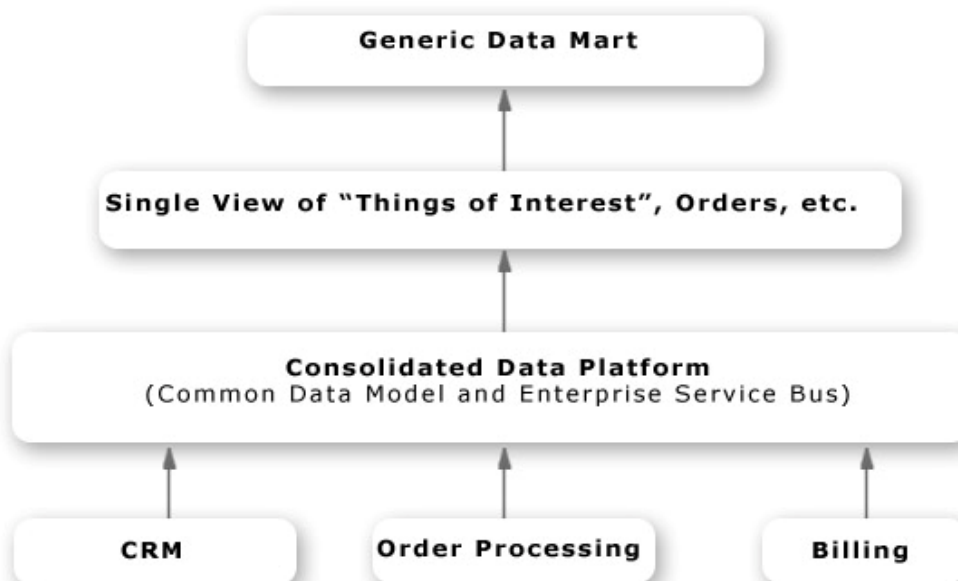
You will also learn the techniques of data management that make this possible and the underlying data models.

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

18.2 The Data Architecture

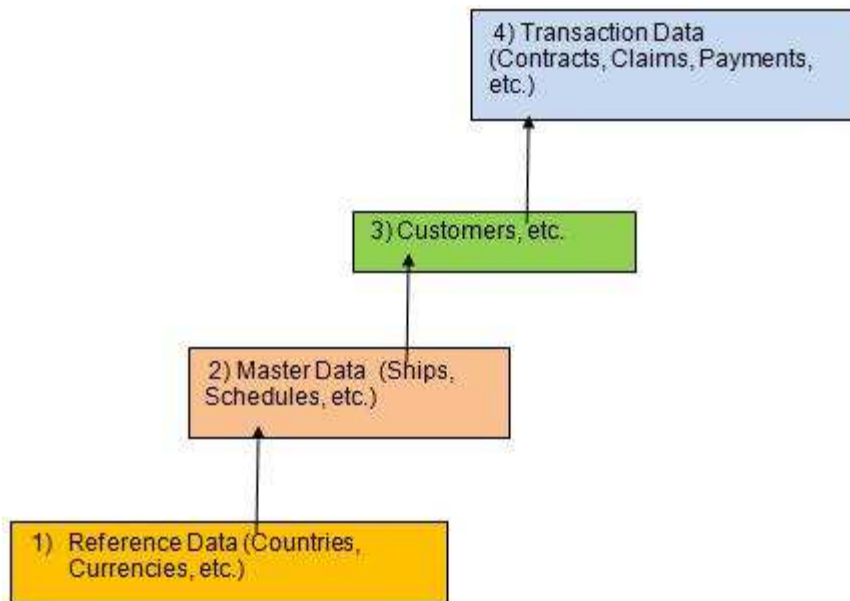
We start with a simple four-layer **data architecture**.

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.



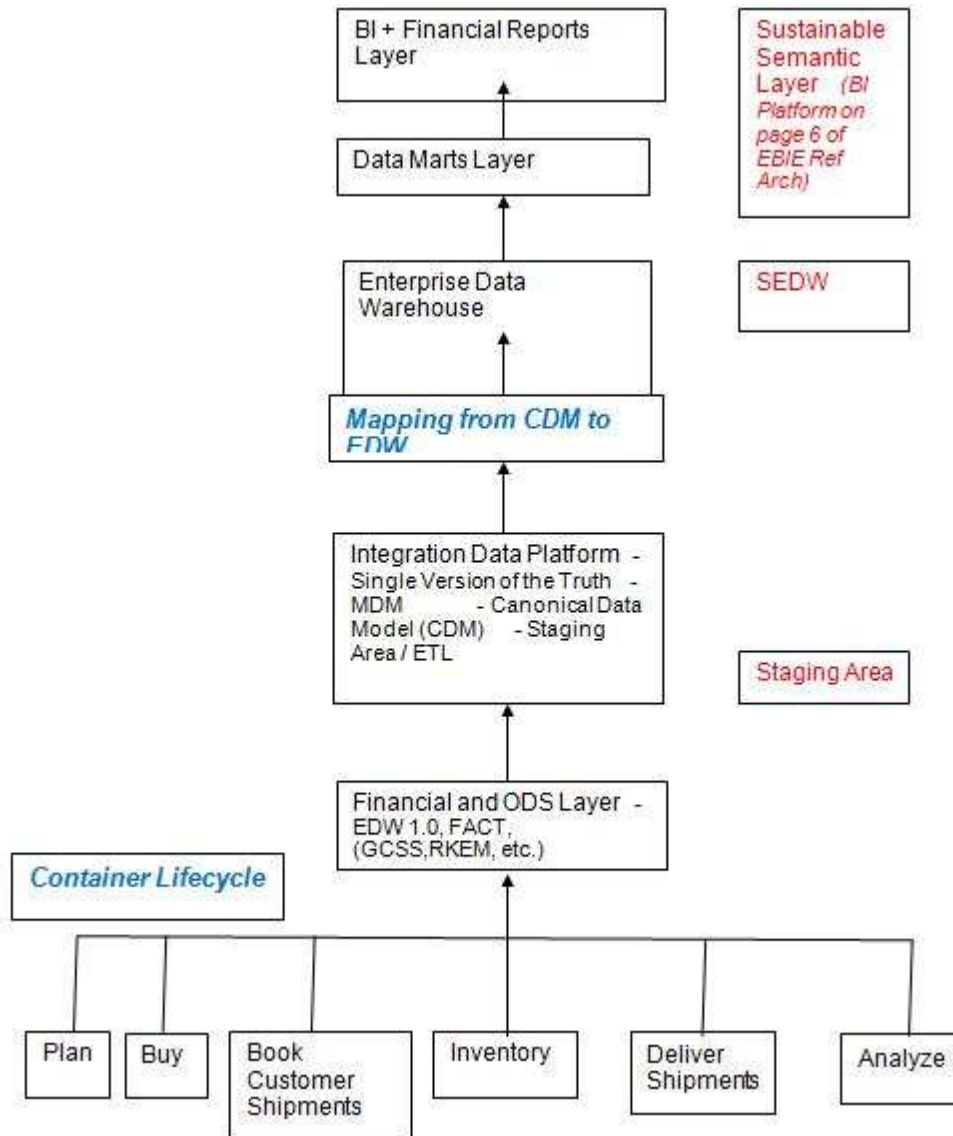
18.3 A Data Loading Sequence

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



18.4 Detailed Data Architecture

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

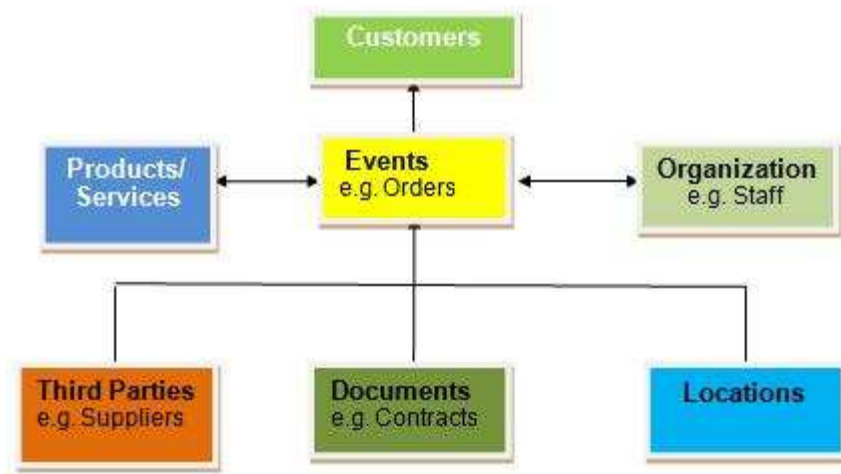


18.5 The Canonical Data Model

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

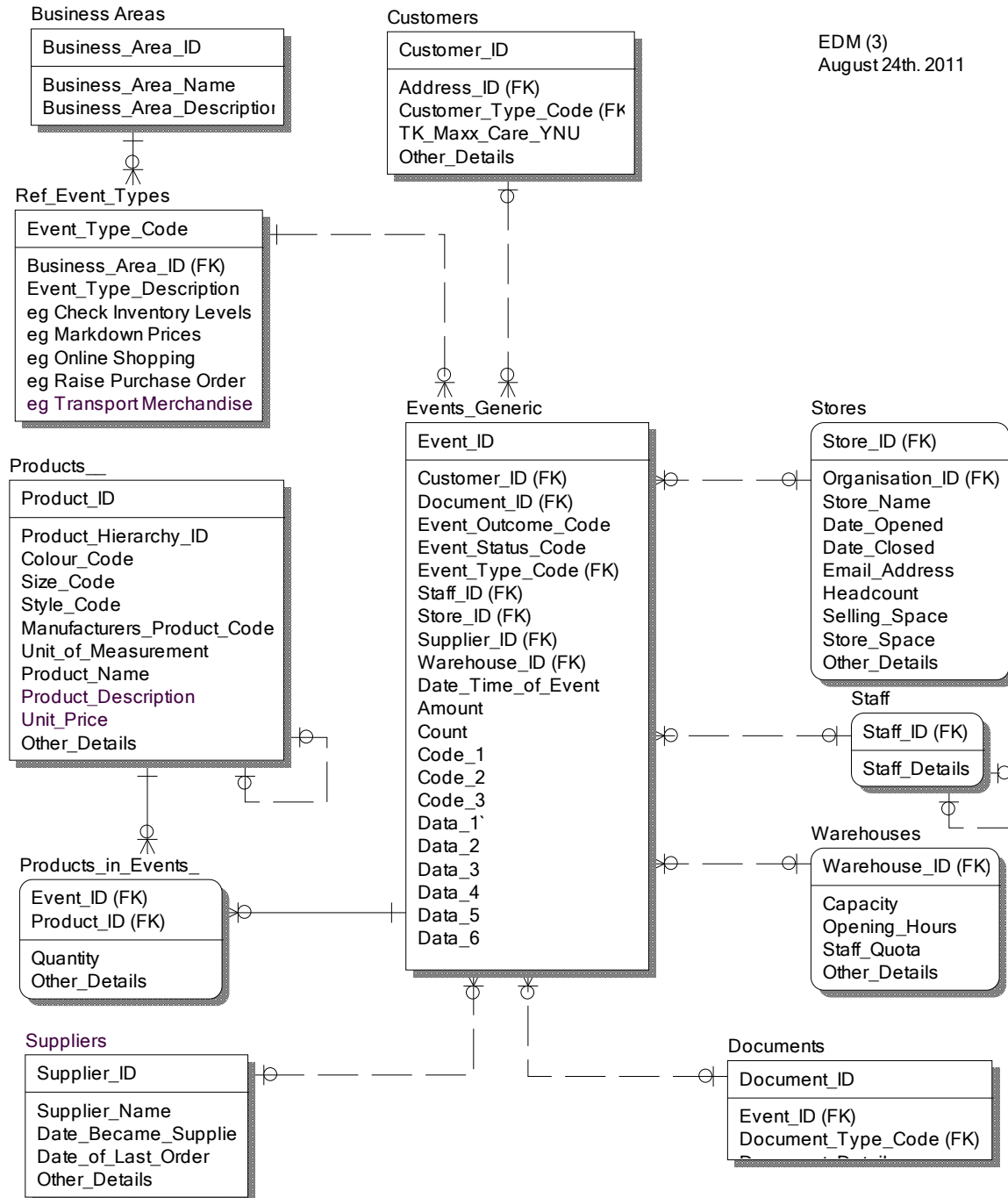
Establishing communication with business users

Transforming data to for loading into data warehouse



18.6 ERD for the Canonical Data Model

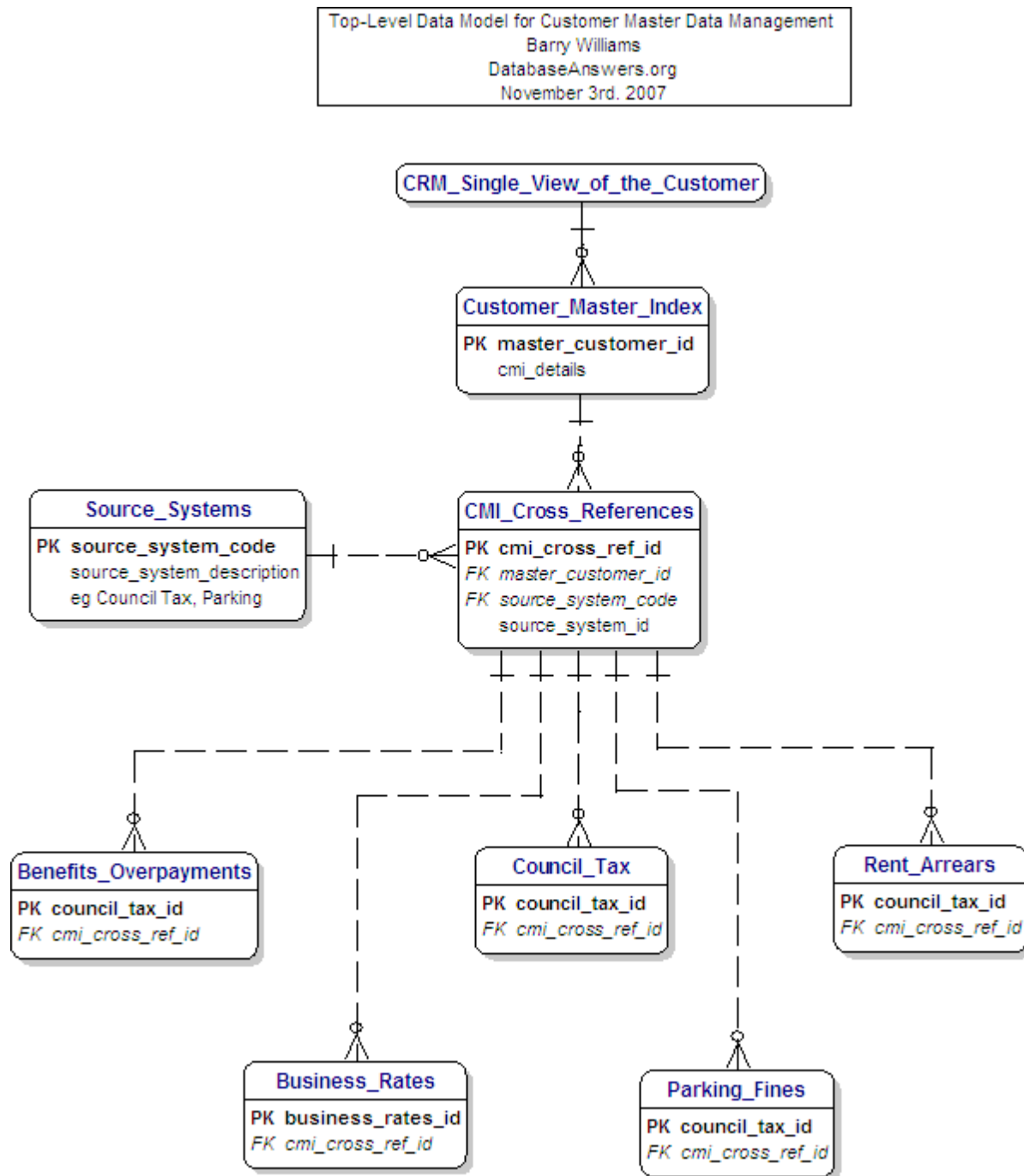
This shows the starting point, which is the generic model for the EDM.



18.7 Master Data Management

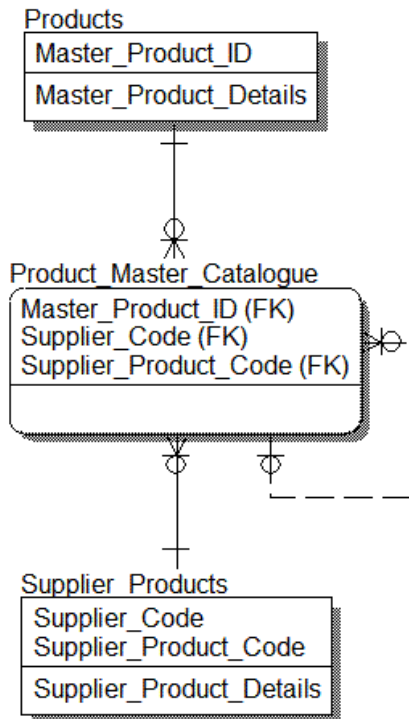
18.7.1 Customer Master Index

This shows a **customer master index** in a local authority environment.



18.7.2 Product Master Catalog

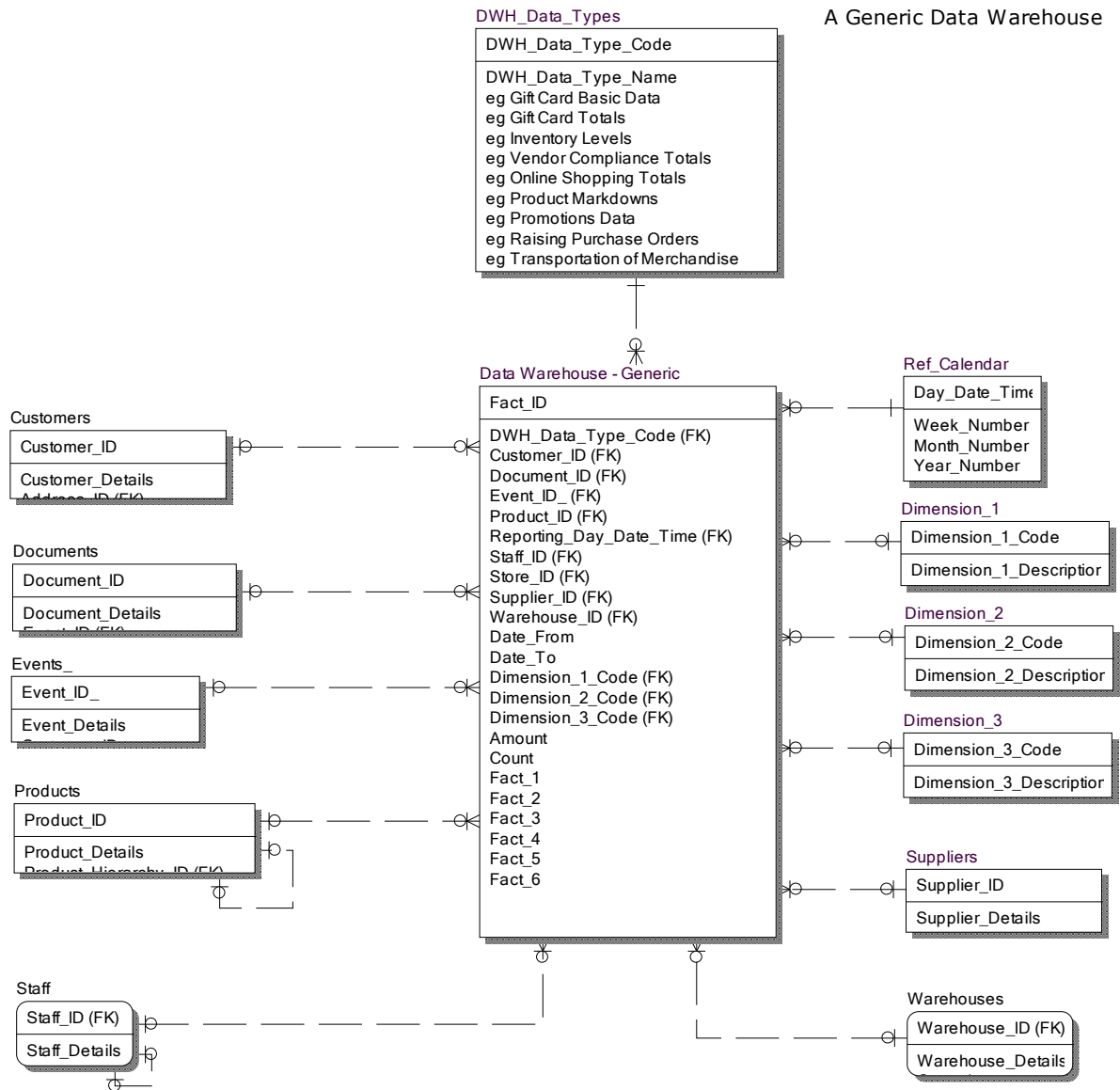
This data model shows that the **product master catalog** plays a central role in matching specific supplier product codes to one unique master product ID.



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

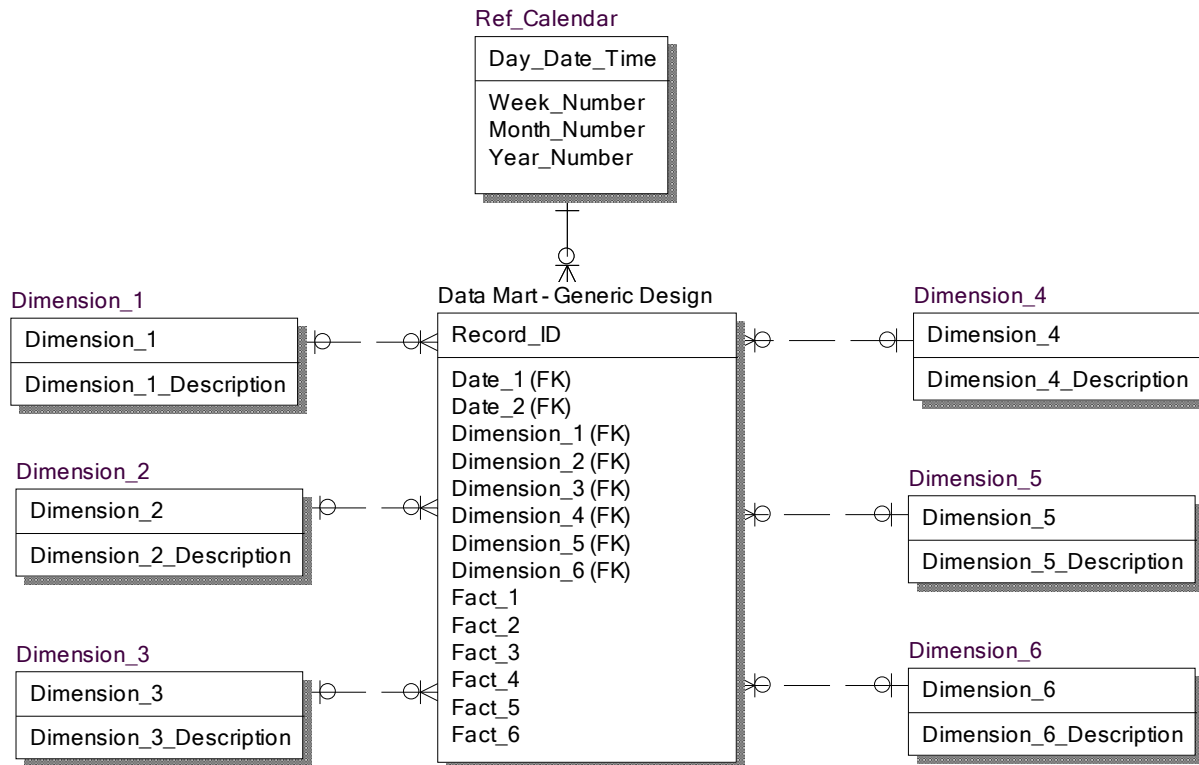
At the logical level, a product or service hierarchy can be defined in a concise manner that offers a high degree of flexibility. This design makes it possible to establish multiple lower level hierarchy chains that belong to the same chain at a higher level. In other words, to have varying degrees of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

18.8 ERD for Generic Data Warehouse



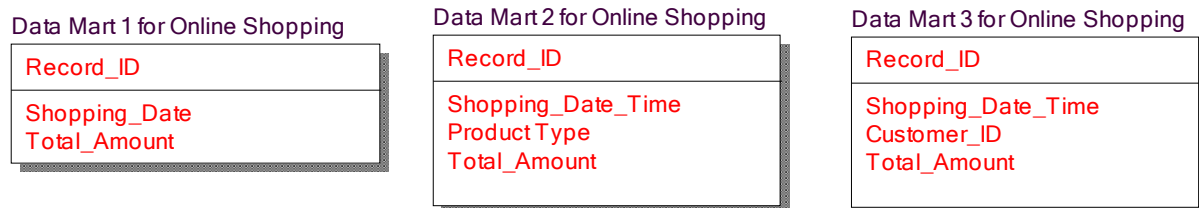
18.9 Generic Retail Data Mart

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.



18.10 Sample Data Marts

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



18.11 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.

19. A Case Study

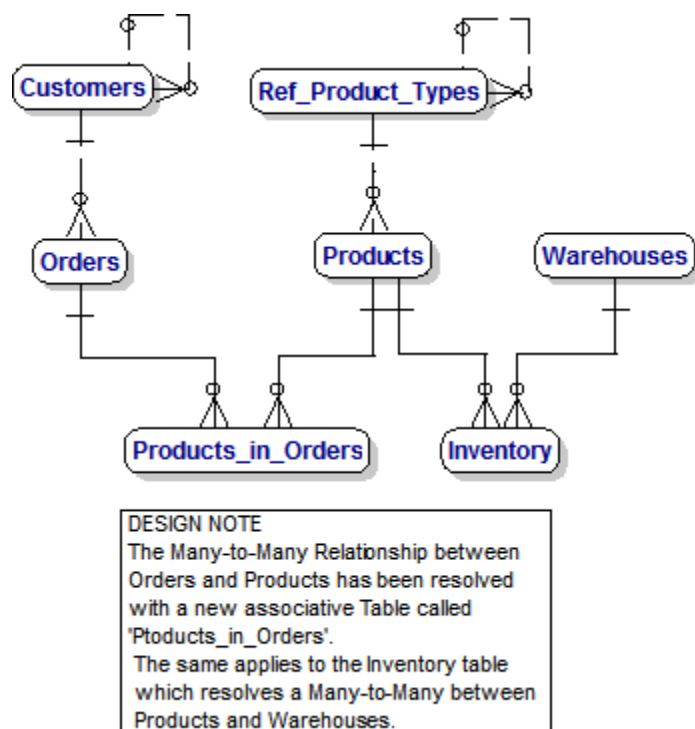
This case study provides an example of the tutorial in action. It includes blank templates and sample templates that are guidelines.

Step 1. Create a Top-Level Business Data Model

19.1 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:



19.2 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:

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

19.3 Functional Description

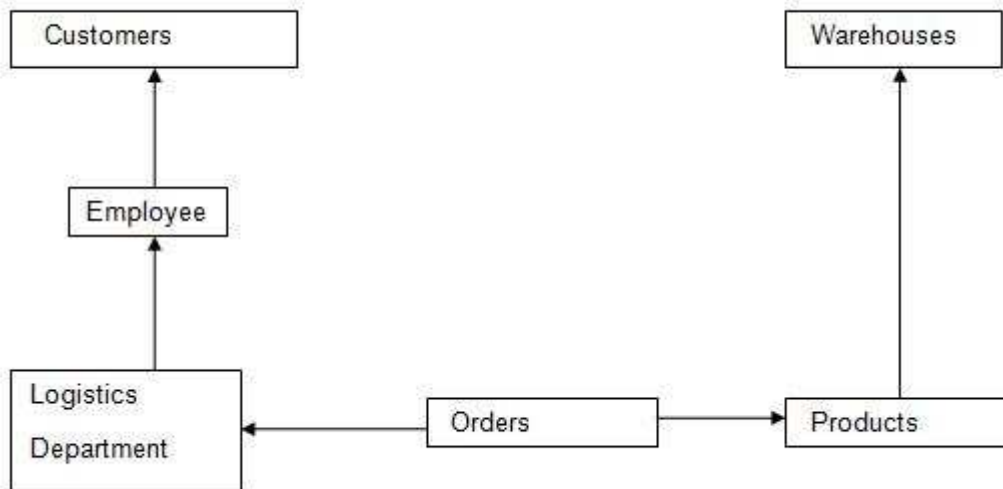
In this diagram, arrows point from children to parents. The scope of the data model includes orders for products from customers.

The Functional description is a simple one-liner:

“Customers issue orders for products that are stored in warehouses”.

19.4 A Specific Model in Word

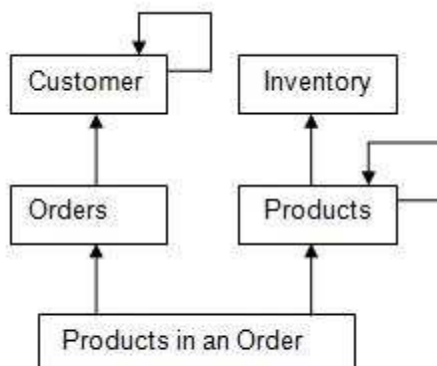
The specific version is consistent with the generic version and looks like this:



19.5 A Generic Model in Word

In this diagram, arrows point from children to parents.

Our generic data mode looks like this:

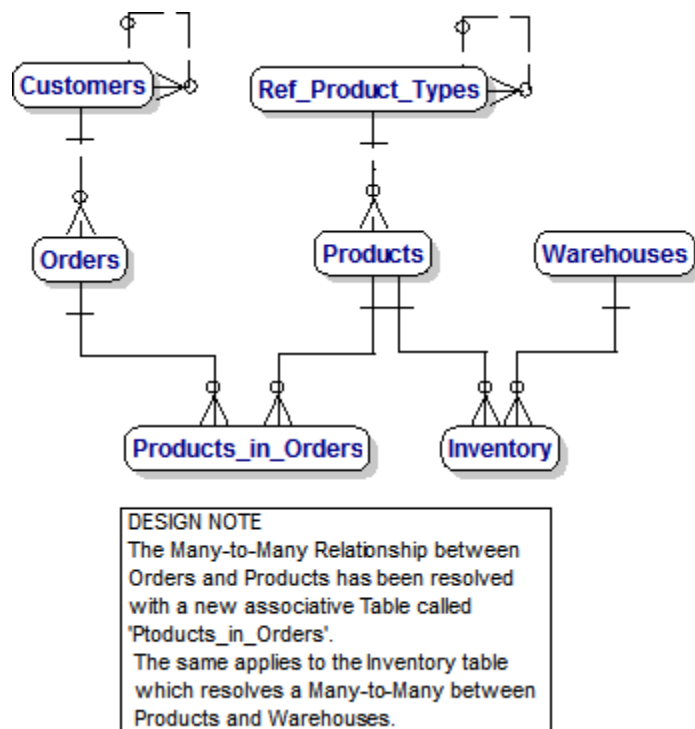


19.6 A Top-Level Generic Data Model

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.



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

Step 2. 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.

A customer is staffed with many employees.

Purchasing departments raise orders.

An order must be authorized by an employee.

An employee is assigned to one department at any point in time.

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 3. 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 4. 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 5. Review with Users

Review and revise as necessary.

Step 6. 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 7. 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 8. 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 11. Check Naming Standards

Standards that are common include:

Initial capitals with lower case elsewhere – for example, Customer_id

All capitals – for example, CUSTOMER_ID

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 9. 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 10. 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 11. Determine the Assurance Level

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

19.7 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/>