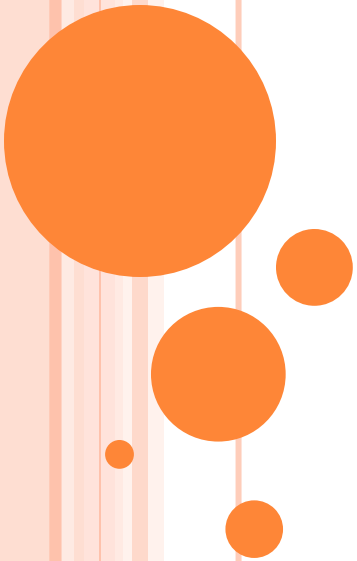


MODELING THE CALENDER

INTRODUCTION

- The calendar is probably the most important, yet least appreciated, component of a data warehouse.
- It serves as the central point of reference for integrating and analyzing business events based on the date.
- The calendar provides important historical perspective for the data warehouse.

Calendars in business

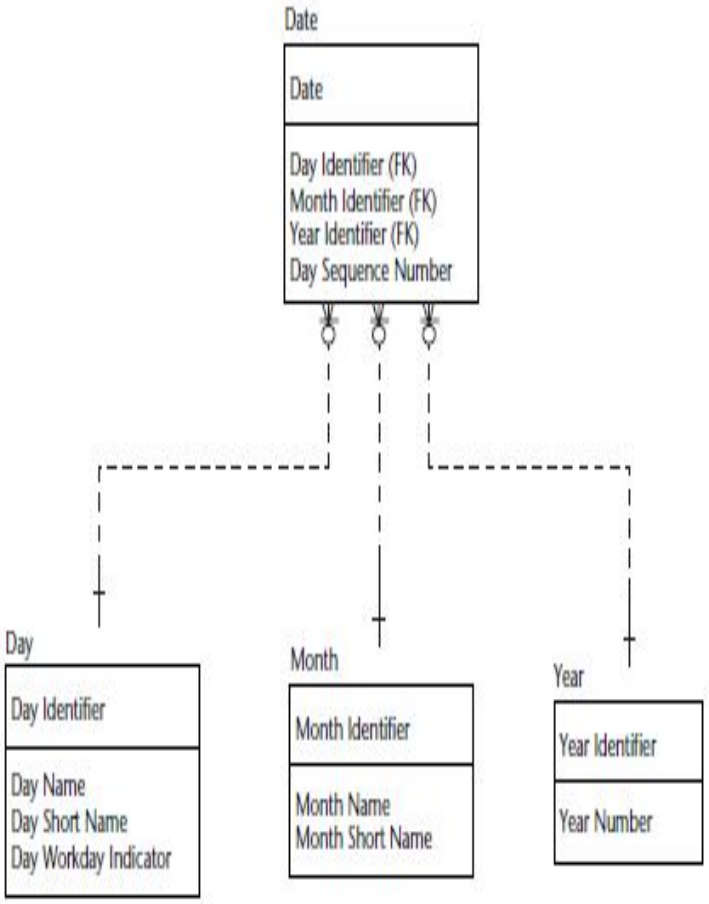
- Businesses use a variety of calendars. The purpose of a calendar is to relate dates for a particular application.
- In business, in addition to the standard (Gregorian) calendar, there's a fiscal calendar that is used for accounting and financial management purposes.
- In addition, some companies have other calendars based on business needs. These include billing-cycle calendars, factory calendars, and others.

Calendars in business

- We will examine the business calendars from these perspectives:
  - the types of calendars,
  - the critical elements within these calendars,

Calendar types

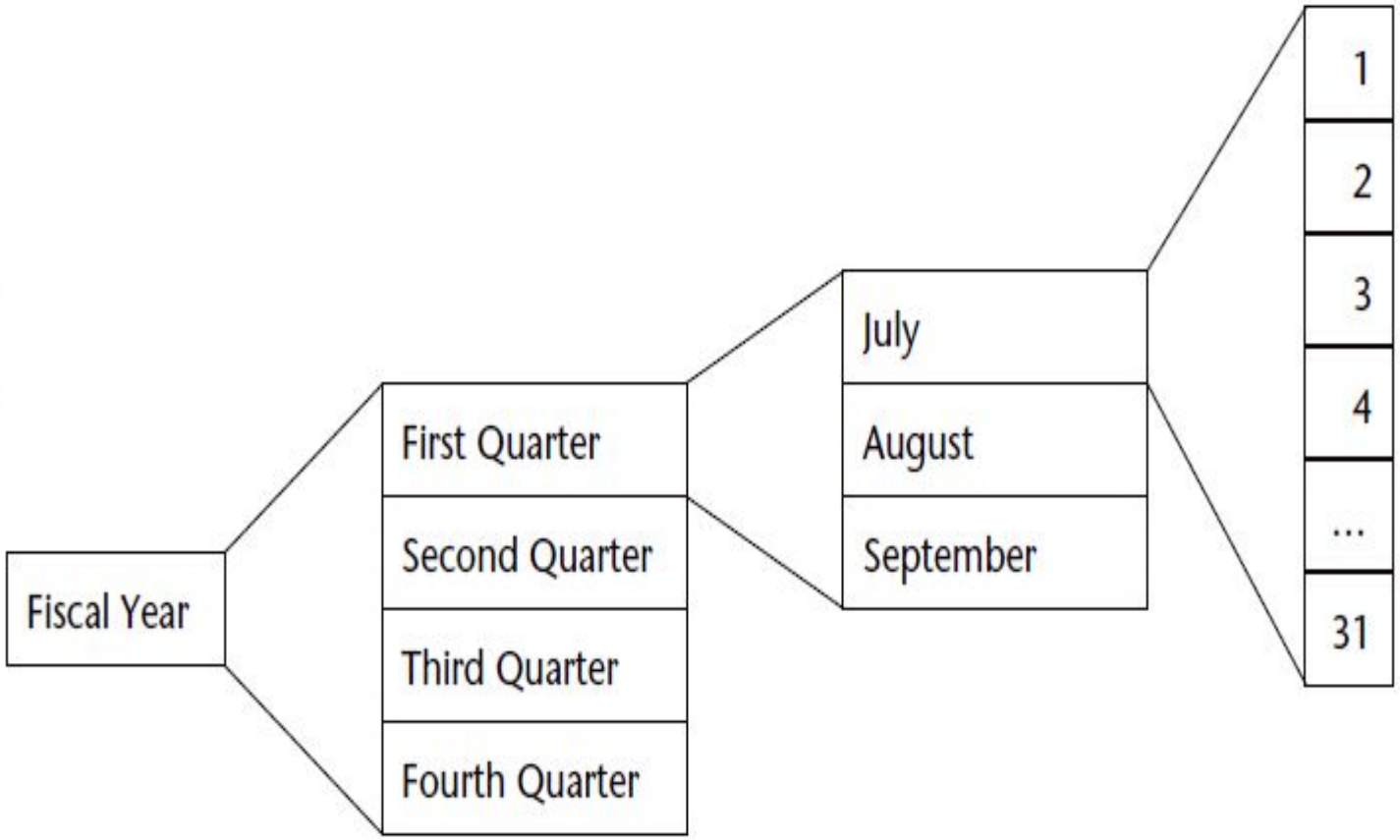
- The **Gregorian calendar** is generally accepted as the basis for establishing dates for business activities.
- Each date is composed of a year, a month, and a day within the month, each through a foreign key relationship.
- In addition, each date can be related to one of the seven days of the week.



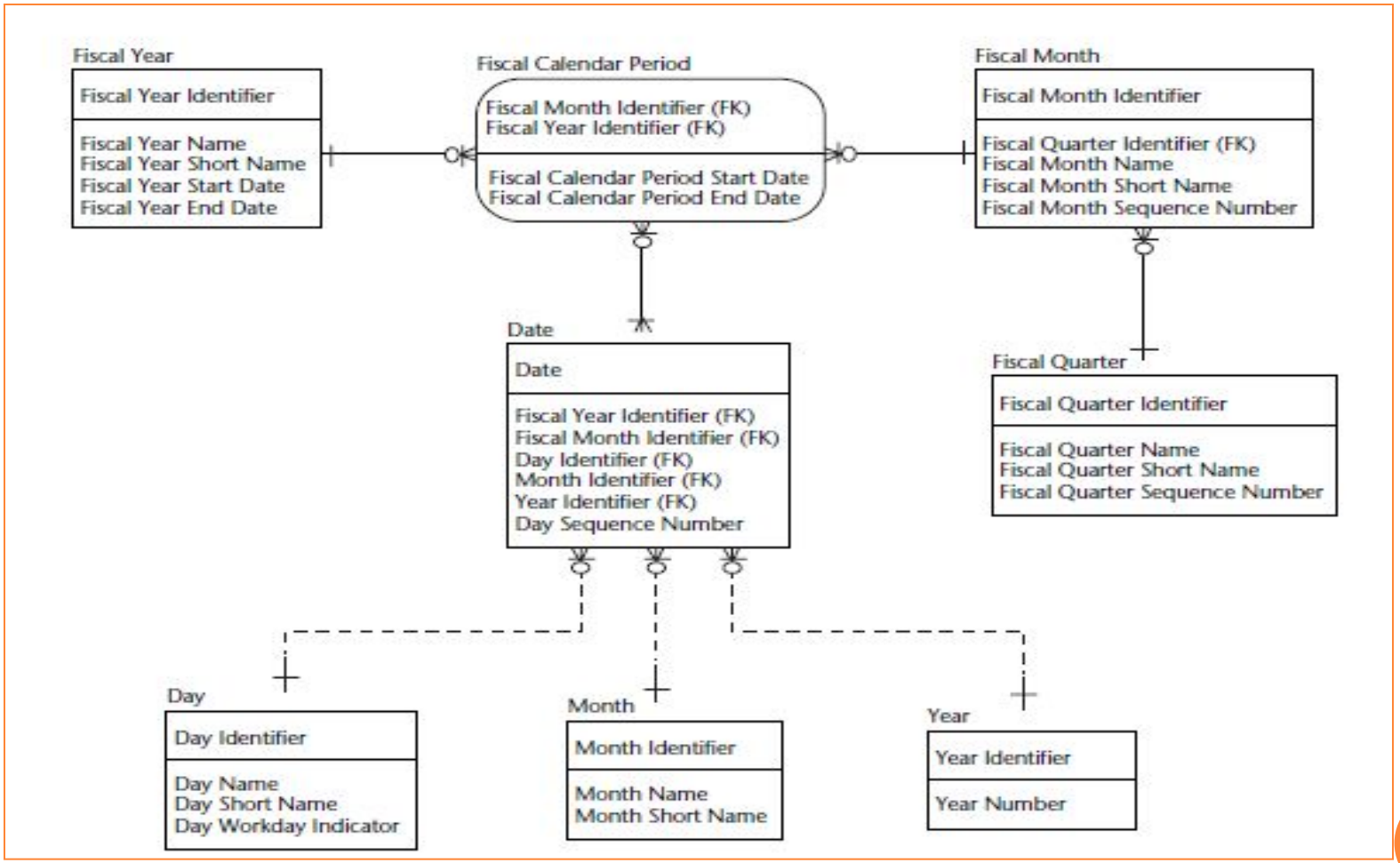
The Fiscal calendar

- The fiscal calendar is the clock that drives financial reporting and accounting practices within the business.
- This calendar is made up of a fiscal year that contains four fiscal quarters, each of which contain three fiscal months, which are sometimes called fiscal periods.
- The fiscal months each contain individual dates, and some fiscal calendars group these into fiscal weeks.

The Fiscal calendar



Fiscal calendar in business data model



The 4-5-4 Fiscal Calendar

- In the retail industry, sales comparisons are often made on a week-to-week basis. When weekly results are rolled up to months, these comparisons lose some meaning because the weeks would need to be split.
- Retail companies have solved this problem by adopting a "4-5-4 calendar" for fiscal analysis and reporting.
- With a potential exception of the year end, this calendar establishes all of its periods based on 7-day periods ending on a particular day of the week, such as Friday.
- Each quarter contains 13 weeks, and these are allocated to the 3 months within the quarter, with the first month containing 4 weeks, the second month containing 5 weeks, and the third month containing 4 weeks.

Other Fiscal Calendars

- There are other fiscal calendars that are unique to a particular business. All of the calendars define a fiscal year, and most define fiscal quarters and fiscal months.
- Regardless of the structure of the fiscal calendar, the data warehouse must support structures that provide both a fiscal and a traditional (for example,
- Gregorian) calendar view of the business's data.

Customer Segment

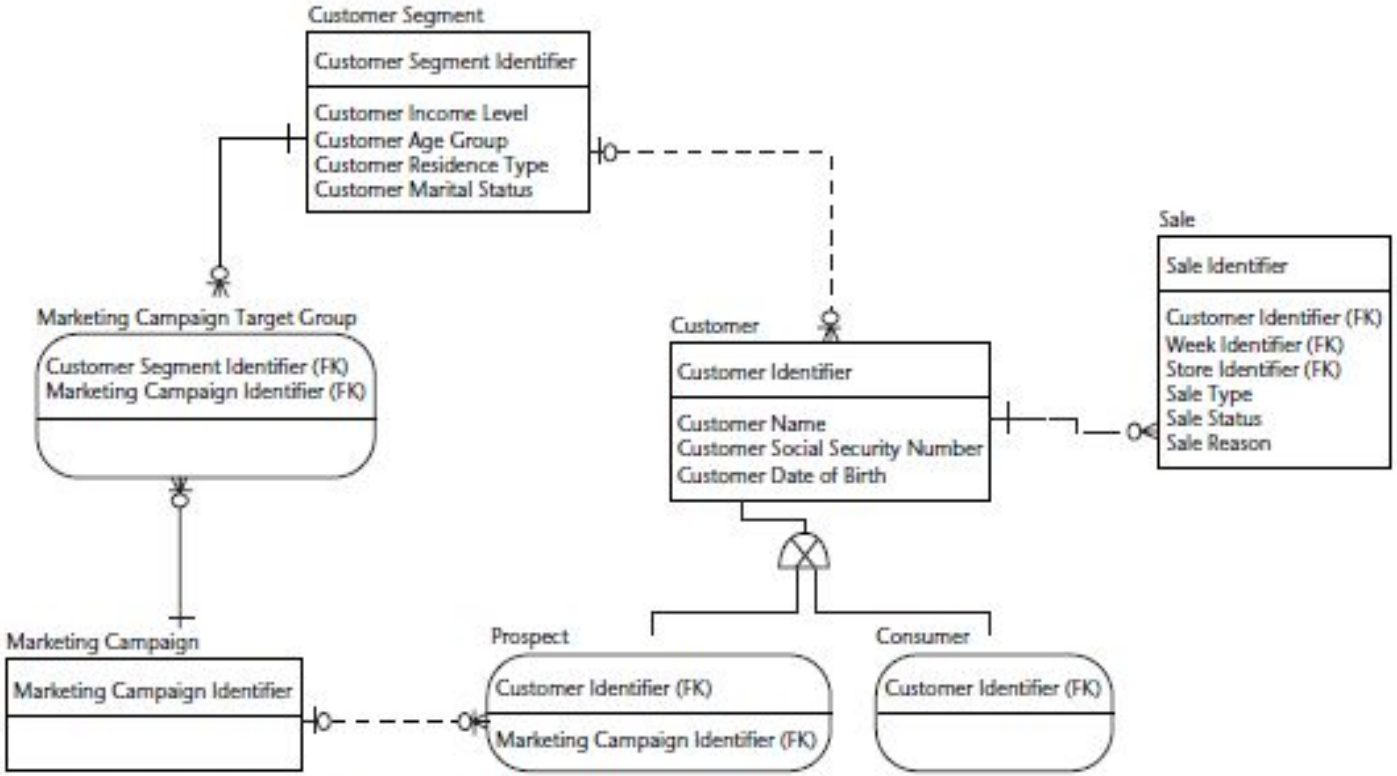
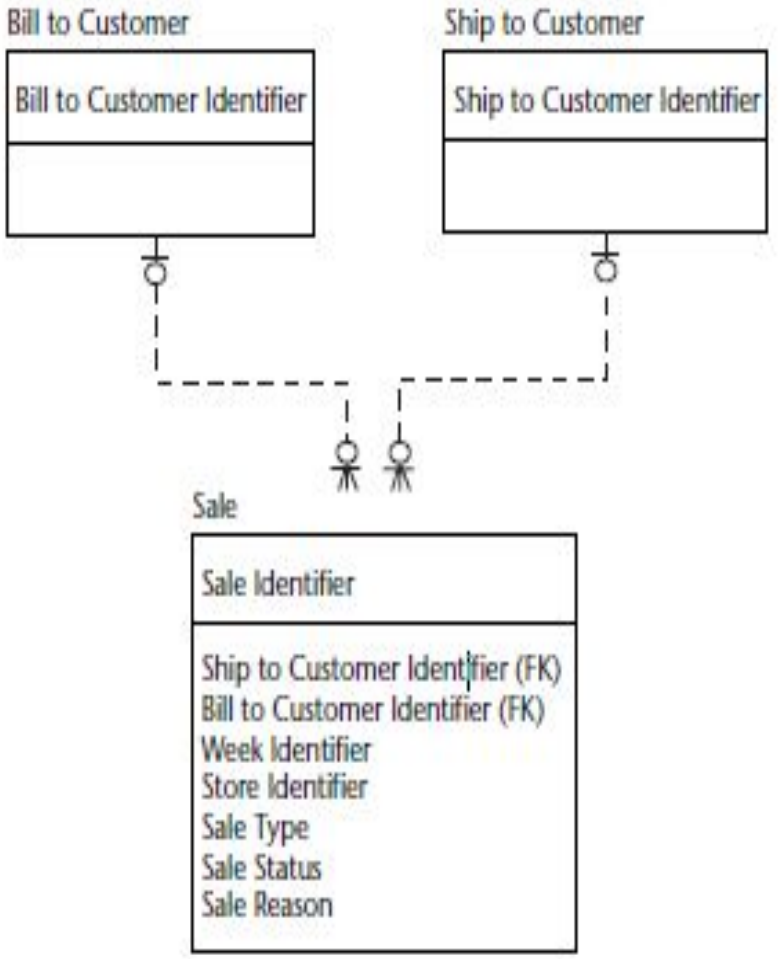


Figure 5.3 Customer segment.

Customers Uniquely Identified Based on Role

- Sometimes, customers in the source system are uniquely identified based on their role.
- For example, the information about one customer who is both a ship-to customer and a bill-to customer may be retained in two tables, with the customer identifiers in these tables being different.

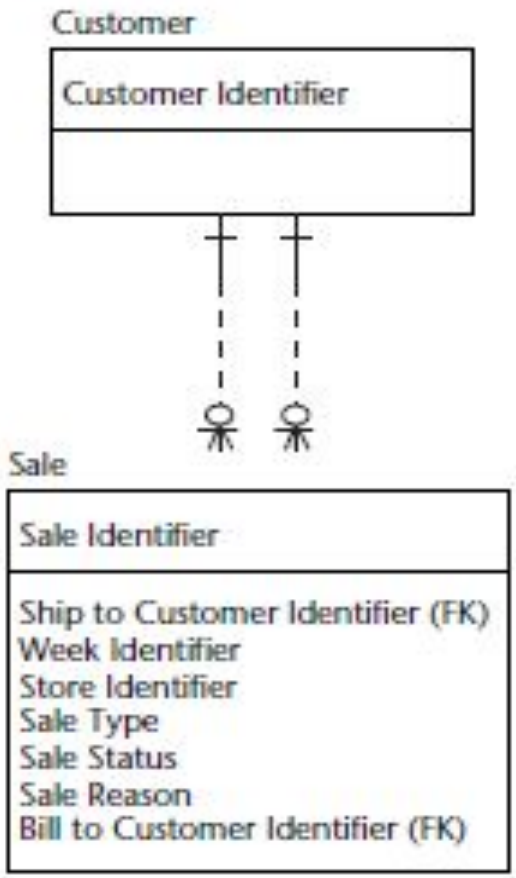


Customers Uniquely Identified Based on Role

- When the tables are structured in that manner, with the identifier for the Ship-to Customer and Bill-to Customer being independently assigned, it is difficult, and potentially impossible, to recognize instances in which the Ship-to Customer and Bill-to Customer are either the same Customer or are related to a common Parent Customer.
- If the enterprise is interested in having information about these relationships, the business data model (and subsequently the data warehouse data model) needs to contain the information about the relationship.

Customers Uniquely Identified Based on Role

- This is typically handled by establishing each role as a subtype of the master entity.
- Once that is done, we reset the identifiers to be independent of the role.
- This results in the relationship shown on the right side of Figure

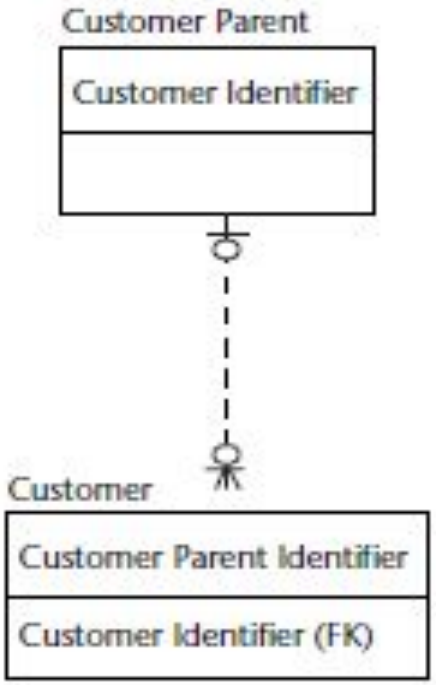


Customer Hierarchy Not Depicted

- Information about customers is not restricted to the company that is directly involved in the sale.
- It is often important to recognize how customers are related to each other so that, if several customers are subsidiaries of one corporation, we have a good understanding of the value of the whole corporation.

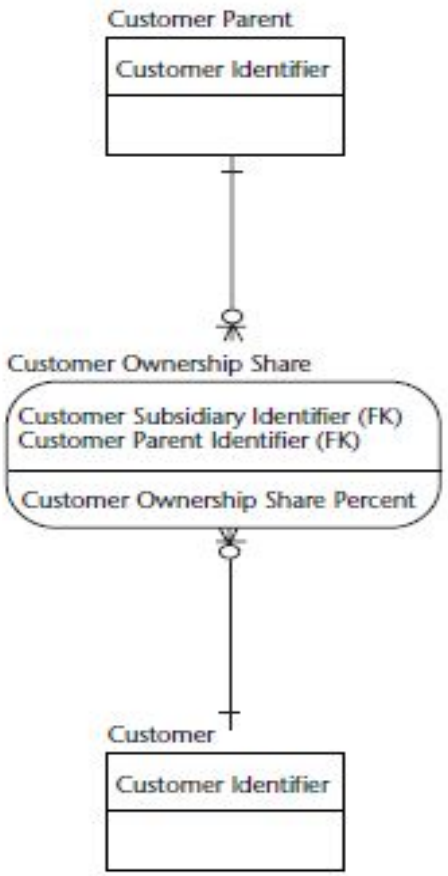
Customer Hierarchy Not Depicted

- Wholly owned subsidiaries are relatively simple to handle since these can be represented by a one-to-many relationship



Customer Hierarchy Not Depicted

- Partially owned subsidiaries are more difficult.
- In this case, the model needs to handle a any-to-many relationship, which is resolved with the associative entity



Data Warehouse System Model

- In the day-to-day running of the company, information is typically viewed in terms of the individual sales transaction life cycle, and the lack of integrated customer information does not prevent making the sale, invoicing the customer, or collecting the payment.
- When the company tries to consolidate all the sales for each of its customers, the lack of integrated customer information becomes a problem.

Data Warehouse System Model

- In the data warehouse, we expect to be able to see all of the sales transactions for each customer, so we need to tackle the problem head on.
- We therefore need to deal with the definitional differences, the lack of unique keys.

Inconsistent Business Definition of Customer

- The inconsistent business definition of Customer was resolved during the creation of the business data model.
- When we build the data warehouse model, we need to select the data elements of interest (Step 1 of the transformation process).

Inconsistent System Definition of Customer

- The data warehouse system model needs to provide a practical target environment for data that is in the operational systems.
- When the definition of the customer differs among systems, then the data warehouse system model needs to be structured so that it can receive all of the legitimate definitions, while maintaining an enterprise perspective of the data.

Inconsistent Customer Identifier among Systems

- Inconsistent customer identifiers among systems cause most of the key integration problems.
- Inconsistent customer identifiers among systems mean that the key structure differs from system to system, and therefore, collecting data for a customer from multiple systems is a challenge.
- A similar problem exists if a system either reuses keys or cannot guarantee that the same customer exists only once in the data file.

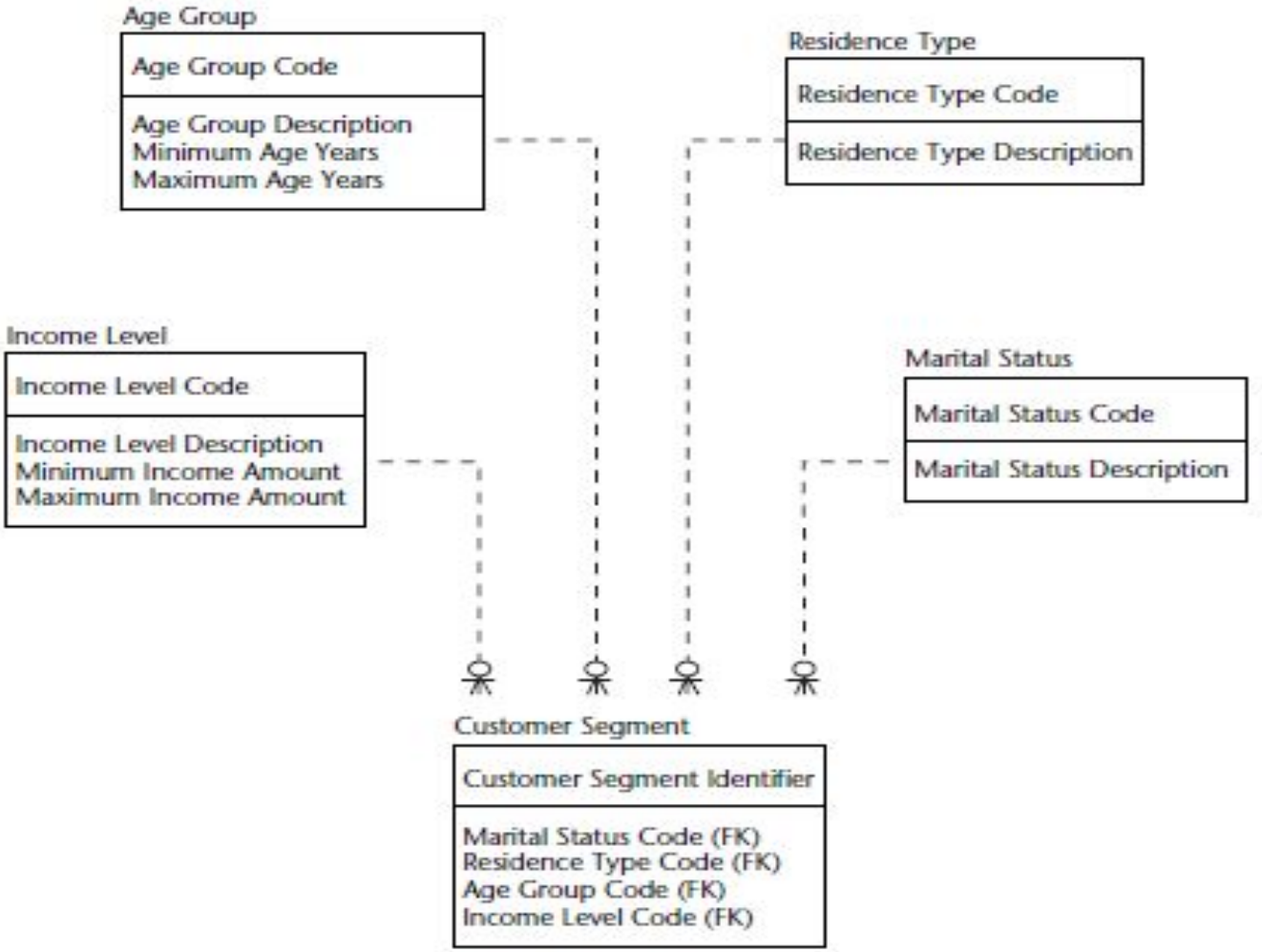
Inconsistent Customer Identifier among Systems

- In the data warehouse model, we simply identify the key as "Customer Identifier."
- In the data warehouse technology model, which transforms into the physical schema, we need to determine how the key is structured.
- When the customer identifiers among the systems vary, most data warehouse modelers lean towards creating a surrogate key.

Absorption of External Data

- For external data based on a set of characteristics of the customer, the business data model includes a single entity representing the Customer Segment.
- Each of the attributes in that entity represents a characteristic, and a discrete set of values exists for each of these.
- Within the data warehouse, these discrete values are typically stored in individual code entities,

Segment Characteristics



Customers Uniquely Identified Based on Role

- The business data model resolved this issue by providing information about the relationship. This resolution is transferred directly to the data warehouse model.

Customer Hierarchy Not Depicted

- The business data model resolved this issue by including the hierarchy (if the number of levels is predictable or consistent) or by deploying a recursive relationship.
- The business model is concerned with a complete picture of the hierarchy. Often, for decision support, we are only interested in specific layers of the customer hierarchy.
- For example, even if we depict four layers in the business data model, the data warehouse model may only need to depict the top and bottom layer.
- Hence, the data warehouse model is more likely to have an exploded structure than a recursive structure.

Data Warehouse Technology Model

- The data warehouse technology model, which is used to generate the physical schema, needs to consider the structure of the key.
- We have three basic options to consider:
  - Use the key from existing system(s)
  - Use the key from a recognized standard
  - Create a surrogate key

Key from the System of Record

- In the simplest situation, we can actually use the key from an existing system.
- For this to be practical, the system must have a key structure that can accommodate data that is derived from other sources.
- This happens when there is one recognized primary source of record for the data, such as an ERP system.
- Some of the needed characteristics of that file follow.
  - **That file should include every customer of interest to the company.**

Key from the System of Record

- **Each customer can exist only once in the file.**
- **The key cannot be reused.**
- **The key is not very long.**
- **The key will not change.**

Key from a Recognized Standard

- There are nationally and internationally recognized code and abbreviation standards. Examples of these include country codes and currency codes.
- Regardless of whether or not any of the systems in the company adopts these standards, the data warehouse can use the standard codes as the key to its code tables.
- The staging area would include a transformation table to translate the code used by the source system to that used in the data warehouse.

Surrogate Key

- A surrogate key is a substitute key. It is usually an arbitrary numeric value assigned by the load process or the database system.
- The advantage of the surrogate key is that it can be structured so that it is always unique throughout the span of integration for the data warehouse.
- When using surrogate primary keys, the term “natural key” is used to refer to the original source system key. The natural key serves as the alternate key for the entity.

Surrogate Key

- Surrogate keys fit all the requirements of a perfect key. They are unique, unambiguous, and never change.
- In addition, surrogate keys provide a number of advantages in the physical database:
  - **The surrogate key is small.**
  - **A surrogate key eliminates compound keys.**
  - **Surrogate keys share the same physical data characteristics.**
  - **A surrogate key is stable.**
  - **The assignment process ensures referential integrity.**

Dimensional Data Mart Implications

- In general, it is most desirable to maintain the same key in the data warehouse and the data marts.
- The data delivery process is simplified, since it does not need to generate keys; drill-through is simplified since the key used in the data mart is used to drill through to the data warehouse.
- However, it is not always possible to maintain the same key structure because of the different techniques used to create the models.