



CRAEATING AND MAINTAINING KEYS

BUSINESS SCENARIO

- Companies endeavouring to implement customer relationship programs have recognized that they need to have a complete view of each of their customers.
- When they attempt to obtain that view, they encounter many difficulties, including:
 - The definition of customer is inconsistent among business units.
 - The definition of customer is inconsistent among the operational systems.
 - The customer's identifier in each of the company's systems is different.

BUSINESS SCENARIO

- ❑ The customer's identifier in the data file bought from an outside party differs from any identifier used in the company's systems.
- ❑ The sold-to customer, bill-to customer, and ship-to customer are separately stored.
- ❑ The customer's subsidiaries are not linked to the parent customer.
- ❑ Each of these situations exists because the company does not have a process in place that uniquely identifies its customers from a business or systems perspective.

INCONSISTENT BUSINESS DEFINITION OF CUSTOMER

BUSINESS UNIT	POTENTIAL DEFINITION	IMPLICATION
Marketing	Any party that might or does buy our product	Includes prospects
Customer Service	A party that owns our product and has an existing service agreement	Includes only customers that we need to support
Sales	Any party that buys our product	This is typically the sold-to or bill-to customer; it excludes the ship-to customer
Commercial Sales	A company that buys our product	Restricted to commercial sales
Manufacturing	Companies that buy directly from us	Excludes retail sales and restricted to commercial sales

INCONSISTENT CUSTOMER IDENTIFIER AMONG SYSTEMS

- ❑ Inconsistent customer identifiers among systems often prevent a company from recognizing that information about **the same customer is stored in multiple places.**
- ❑ This is not a business data modelling issue—it is a data integration issue that affects the data warehouse data model,

INCONSISTENT SYSTEM DEFINITION OF CUSTOMER

- Operational system was developed without the use of a governing business model. Any operational system that applies sound data management techniques and applies a business model to its design will be consistent with the business data model.
- Second, differences could exist because of special circumstances that need to be handled. For example, the system changed to meet a business need, but the definitions were not updated to reflect the changes.

INCONSISTENT SYSTEM DEFINITION OF CUSTOMER

- The third reason this situation could exist is that a programmer did not fully understand the overall system design and chose an approach for a system change that was inappropriate.

INCLUSION OF EXTERNAL DATA

- Companies often need to import external data. Examples include **credit-rating** information used to assess the risk of providing a customer with credit, and **demographic information** to be used in planning marketing campaigns.
- There are two basic types of external data relating to customers:
 - data that is at a customer level, and
 - data that is grouped by a set of characteristics of the customers.

DATA AT A CUSTOMER LEVEL

- Integrating external data collected at the customer level is similar to integrating data from any internal operational source.
- The problem is still one of merging customer information that is identified inconsistently across the source systems.
- In the case of external data, we're also faced with another challenge—the data we receive may pertain to more than just our customers (for example, it may apply to all buyers of a particular type of product), and not all of our customers are included (for example, it may include sales in only one of our regions).

DATA GROUPED BY CUSTOMER CHARACTERISTICS

- External data is sometimes collected based on customer characteristics rather than individual customers. For example, we may receive information based on the age, income level, marital status, postal code, and residence type of customers.
- Common approach for handling this is to create a Customer Segment entity that is related to the Customer

CUSTOMER SEGMENT

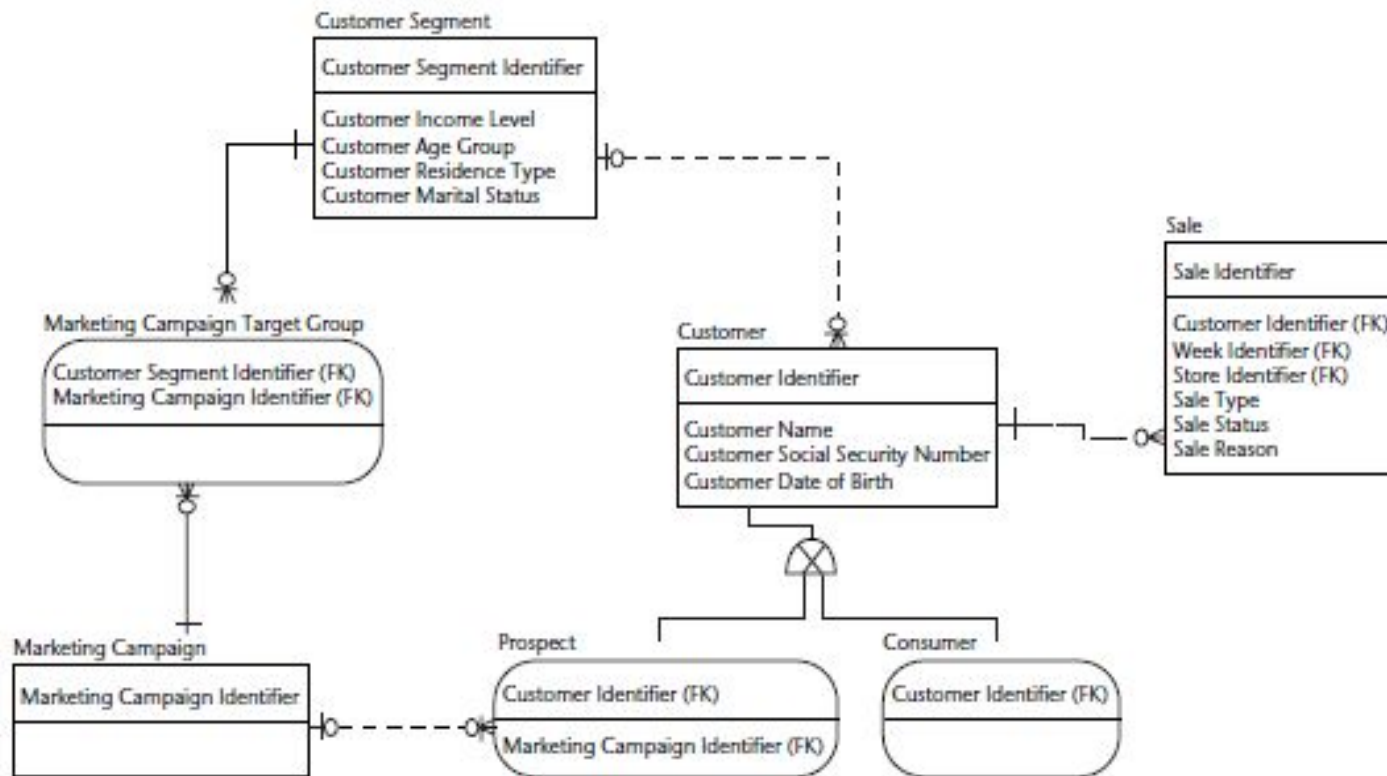
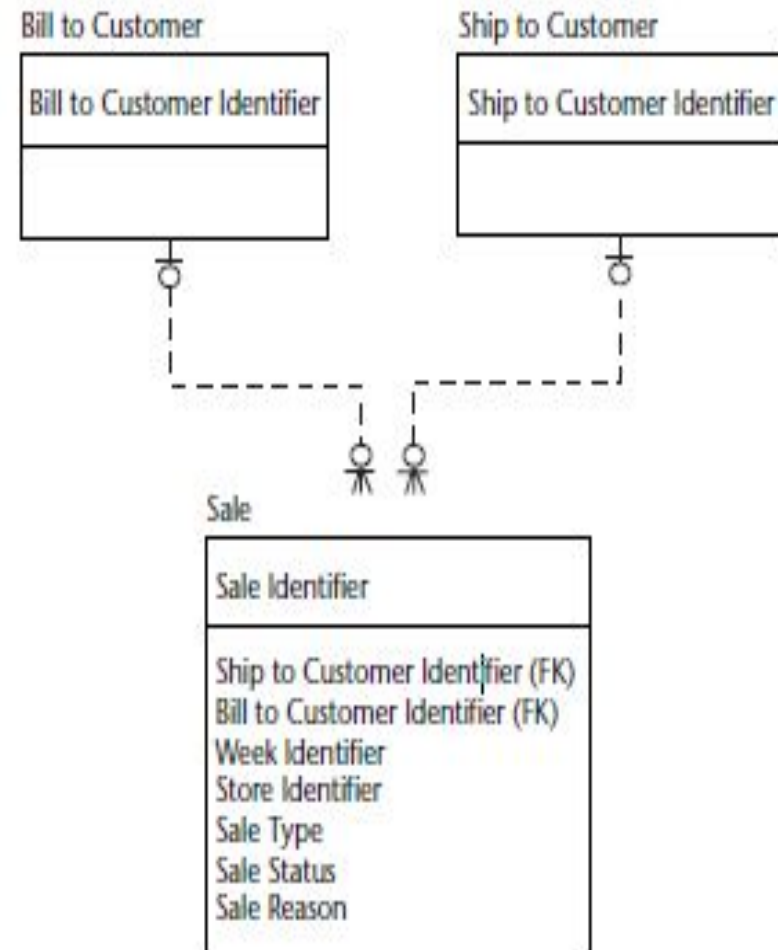


Figure 5.3 Customer segment.

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.

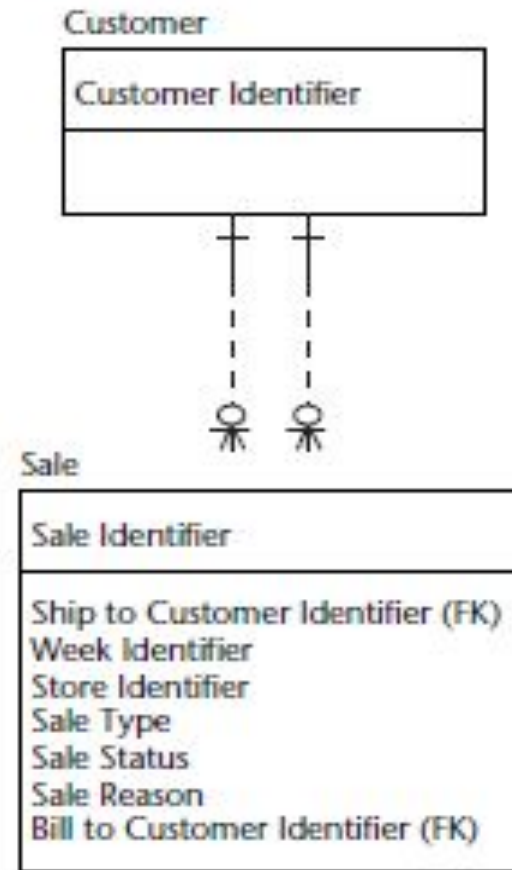


CUSTOMERS UNIQUELY IDENTIFIED BASED ON ROLE

- 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

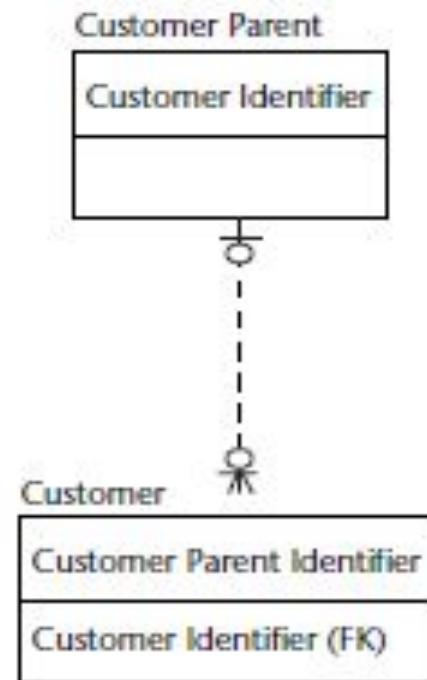


CUSTOMER HIERARCHY NOT DEPICTED

- 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.

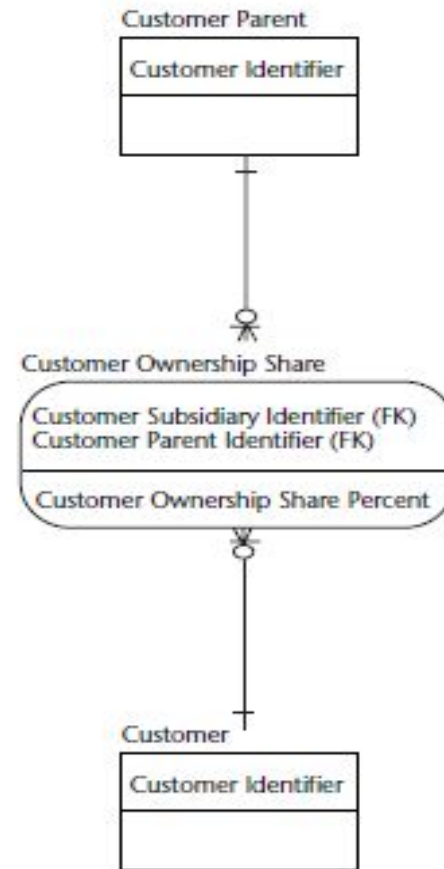
CUSTOMER HIERARCHY NOT DEPICTED

- 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.

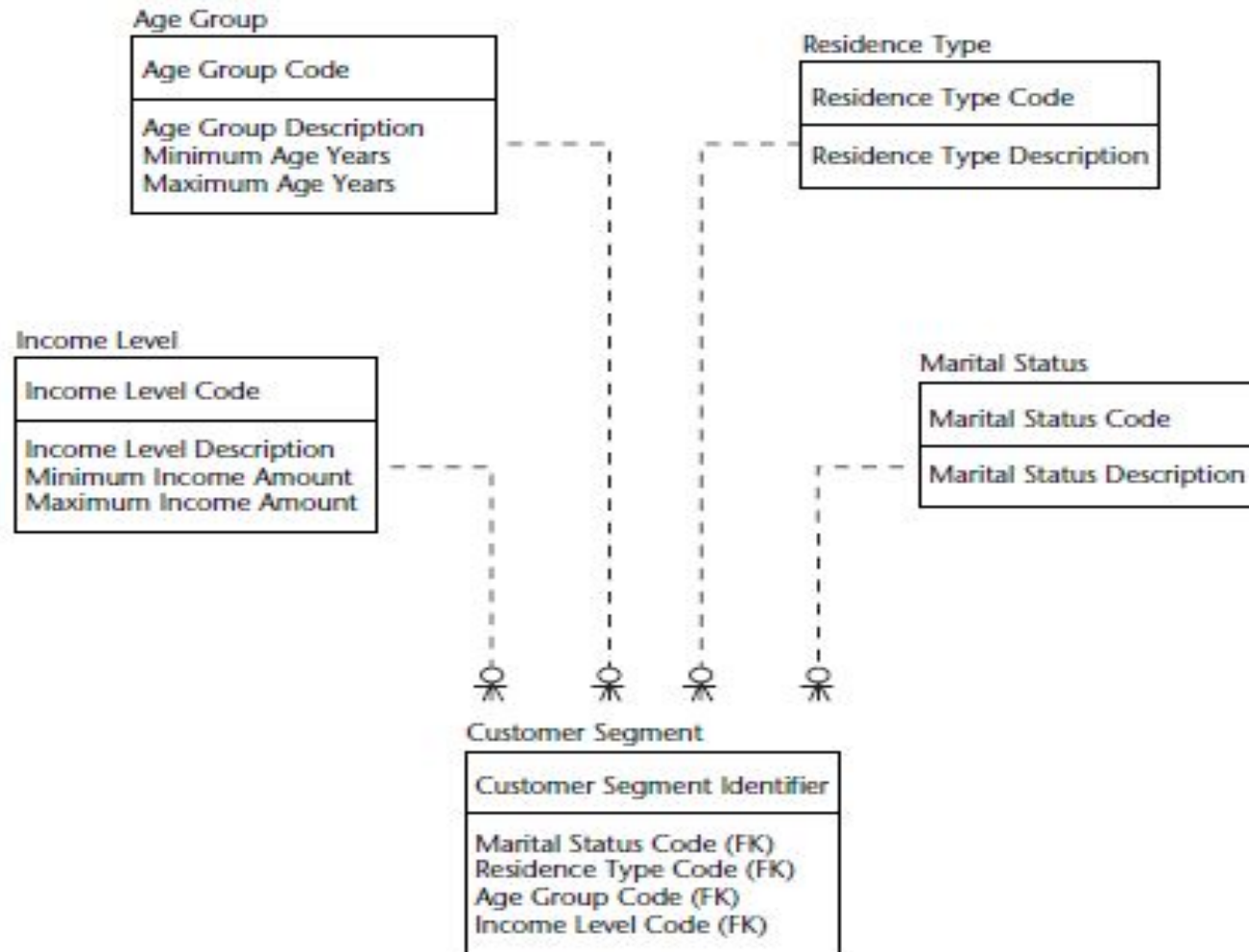
INCONSISTENT CUSTOMER IDENTIFIER AMONG SYSTEMS

- ❑ 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.