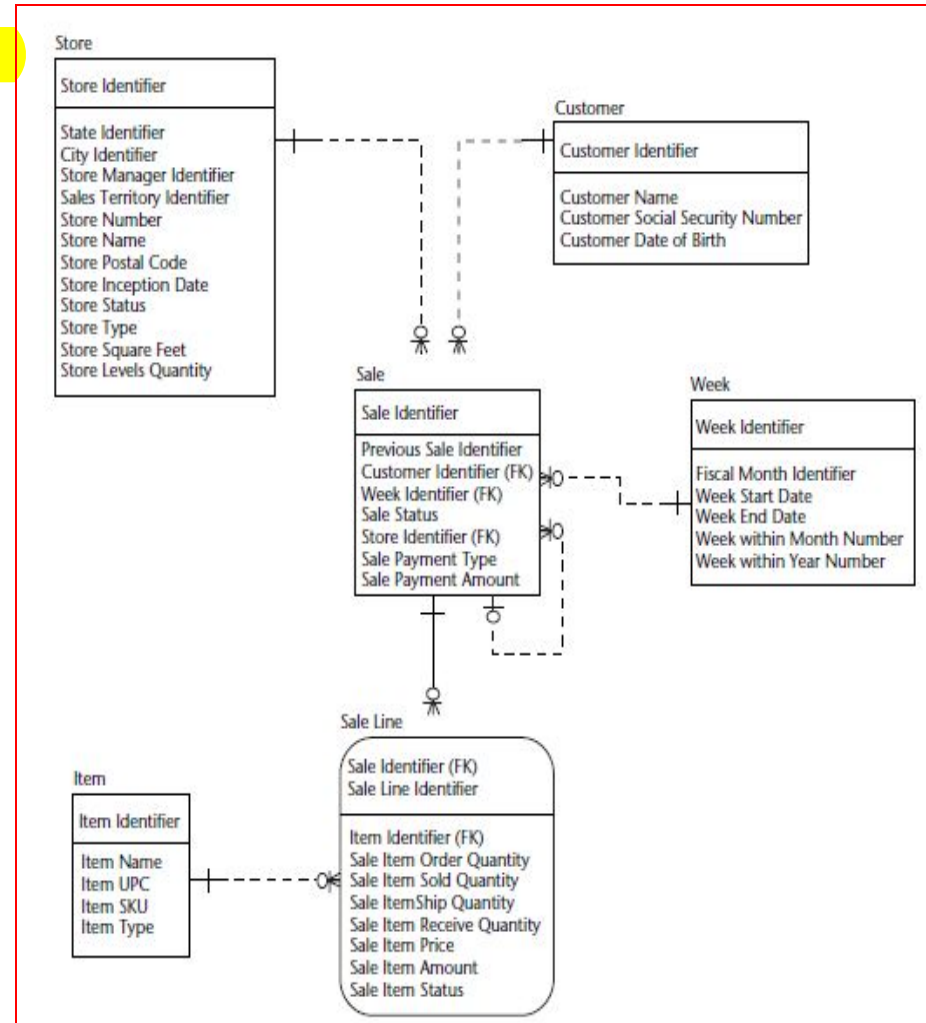


# MODELING TRANSACTIONS

# BUSINESS TRANSACTIONS

- Transaction is a single activity, such as the delivery of an item to a customer in exchange for payment as shown in Fig
- This sales transaction is a one-time event, and once transacted, does not change.



## FACTORS THAT MAKE TRANSACTION COMPLEX

- Payment could have been made using a company credit card, and the business is interested in tracking information about the sale until the payment is actually received.
- The customer paid for part of the purchase in cash and part with a credit card.
- The customer purchased a large number of items that need to be delivered over time. The transaction remains open until delivery is complete.
- The customer could log onto the company's Web site and order the item, so the transaction actually begins with the order.

# BUSINESS USE OF THE DATA WAREHOUSE

- The anticipated business use of the data warehouse is a major driver of its design.
- The first step is to select the data elements of interest. In the case of the transactional data, this step is pursued in two parts:
  - First, we must determine which states of the order are of interest to us. A data warehouse that is built for sales analysis may only be interested in completed sales, and hence may not require information from open orders.

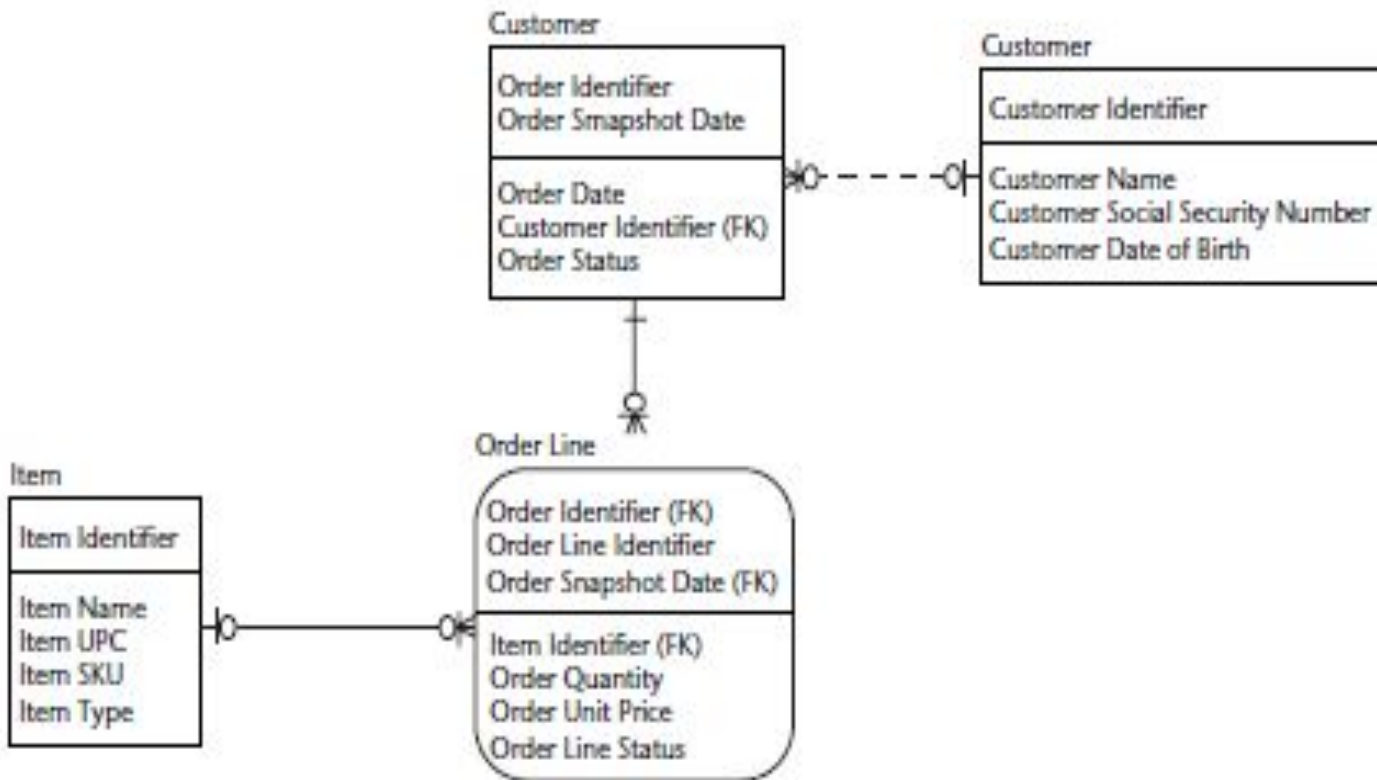
# BUSINESS USE OF THE DATA WAREHOUSE

- A data warehouse that is built to provide information for forecasting or logistics, on the other hand, may require open order information.
- A data warehouse that is built to provide information for forecasting or logistics, on the other hand, may require open order information. Once we decide upon the states of interest, we must then examine each data element to determine the elements that are of interest.

# BUSINESS USE OF THE DATA WAREHOUSE

- The second step is to ensure that we maintain the historical perspective. If we're only interested in closed sales, and a transaction can never change after it is closed, then there is no need for an additional time variant component of the key since a closed transaction can have only one occurrence.

# BUSINESS USE OF THE DATA WAREHOUSE



**Figure 8.2** Open order over time.

# BUSINESS USE OF THE DATA WAREHOUSE

- The third step is to add derived data. Each sales detail line may have a quantity of an item and a price for that item. The extended cost represents the product of the two.
- Discounts may be applied to a sale, and sometimes, they are apportioned among the various items sold.
- If the business rule for “net sales amount” is that it is the revenue derived from selling the item after considering discounts, and excluding taxes, then a derived field could be included in the data warehouse.



# BUSINESS USE OF THE DATA WAREHOUSE

- The fourth step is to ensure that we capture the appropriate level of granularity.
- This is the point at which we decide whether or not to actually store the transaction in the data warehouse. Storing each transaction is expensive, and it is worth doing only if transactional level information is needed for analysis.
- If the analysis only requires data that is summarized (daily sales of each item at each store), then the transaction-level data need not be brought into the data warehouse.

# APPLICATION INTERFACES

- We divide the act of moving data into three parts: the interface, the load, and the data warehouse.
- The interface refers to all aspects of extracting data from a source system and delivering it to the data warehouse load process.
- It encompasses the data, its format, and the process that delivers it.
- The load refers to the processes that transform and load the data received from the interface into the data warehouse.
- Finally, the data warehouse refers to both the model and physical instantiation of the database.

## CATEGORIES OF APPLICATION INTERFACES

- There are two general categories of application interfaces:
  - snapshots and
  - delta interfaces.
- A snapshot interface presents the data as it exists at the time of extraction.
- A delta interface contains changes to the data since the last time data was extracted.

# SNAPSHOT INTERFACES

- A snapshot interface provides a picture of what the data looks like at the time it was extracted. It provides no information about what occurred in the system between the time of the last snapshot and the new one.
- Snapshots are commonly used for reference data, such as code description tables, and may also be used for larger datasets, such as customer or product data.
- They are also used to extract complex structures, such as recursive hierarchy trees, and for infrequent interfaces, such as a monthly inventory extract.

## COMPLETE SNAPSHOT INTERFACE

- Complete snapshot interface is simply an extraction of everything that exists in a table or tables within the application database, sometimes without any discrimination as to the currency or validity of the data. This is a snapshot in its purest form.

## CURRENT SNAPSHOT INTERFACE

- This type of interface delivers data that represents the current active state of the data.
- Obsolete and deleted items do not appear in the current snapshot interface.
- This is often done to reduce the volume of data delivered or where obsolete data is not available from the application system.

# DELTA INTERFACES

- Delta interfaces get their name from the triangular Greek symbol that is used in mathematics to mean *change in*. Thus, a delta interface is one that provides changes to existing data.
- Transactional data is almost always delivered in some form of delta interface. In all cases, a delta interface is aware of the last time data was extracted.
- The extract will always contain what has changed since the last extract or the impact of that change.

## COLUMNAR DELTA INTERFACE

- The columnar delta interface, also referred to as a change log, is the most detailed and voluminous of delta interfaces.
- Such an interface contains one row per column changed. The row would contain the primary key of the data that was updated, the name of the column that was changed, a timestamp for when the change occurred, and the before and after values for the column.



## ROW DELTA INTERFACE

- A row delta interface is similar to the columnar delta interface, except that instead of one row per column, it contains a single row for all columns.
- In this type of interface, the row is populated with the primary key and any attribute values that have changed.
- Attributes that did not change would be null or contain some predefined value indicating no change.

## DELTA SNAPSHOT INTERFACE

- The delta snapshot is a commonly used interface for reference data, such as a customer master list.
- The basic delta snapshot would contain a row or transaction that changed since the last extraction.
- It would contain the current state of all attributes without information about what, in particular, had changed.

# DATABASE TRANSACTION LOGS

- Database transaction logs are another form of delta interface. These transaction logs are maintained by the database system itself at the physical database structure level to provide restart and recovery capabilities.
- There are three main challenges when working with database logs.
  - The first is reading the log itself.
  - The second challenge is applying a business context to the content of the logs.
  - The third challenge is dealing with software changes in both the application system and the database system.