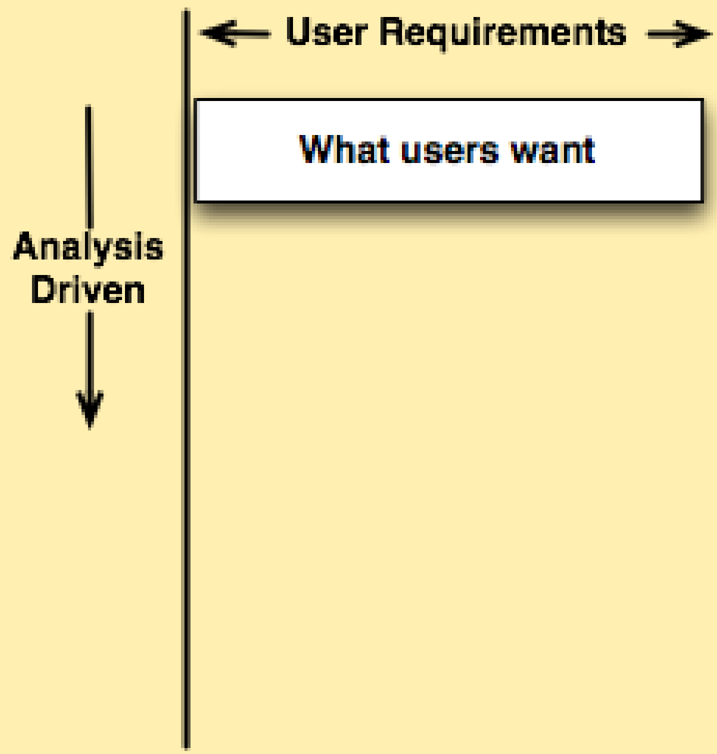
DW CONCEPTUAL DESIGN APPROACHES

Analysis Driven (Bottom-Up, Metric Pull) Design - Kimball

A separate data mart is designed for each business process, and later these schemas are merged forming a coherent global schema for the entire DW. This approach has a limited cost and delivery time.

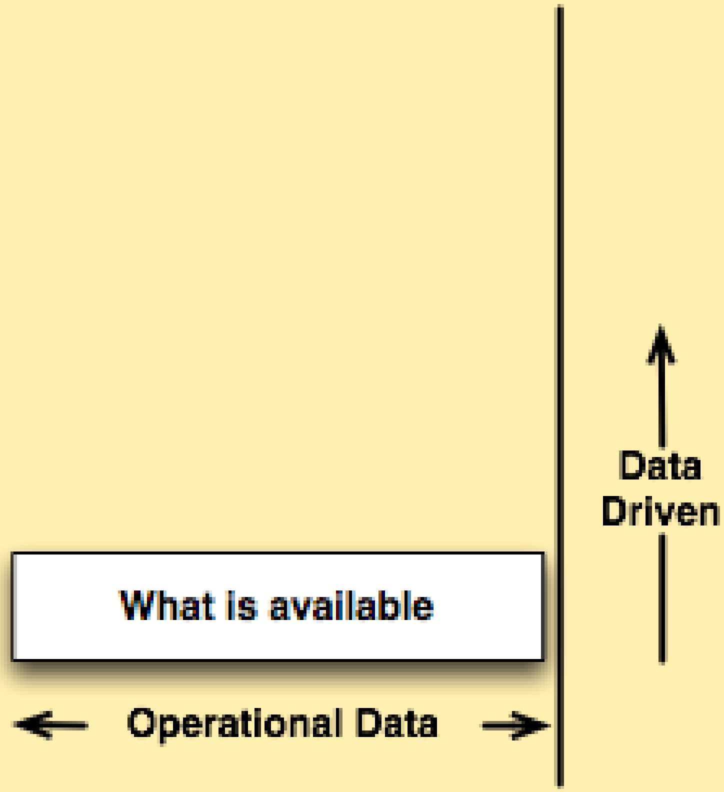


DW CONCEPTUAL DESIGN APPROACHES

In most cases the potential users do not understand what the BI tools could be used for when they first see them, so there could not be any significant demand for BI applications.

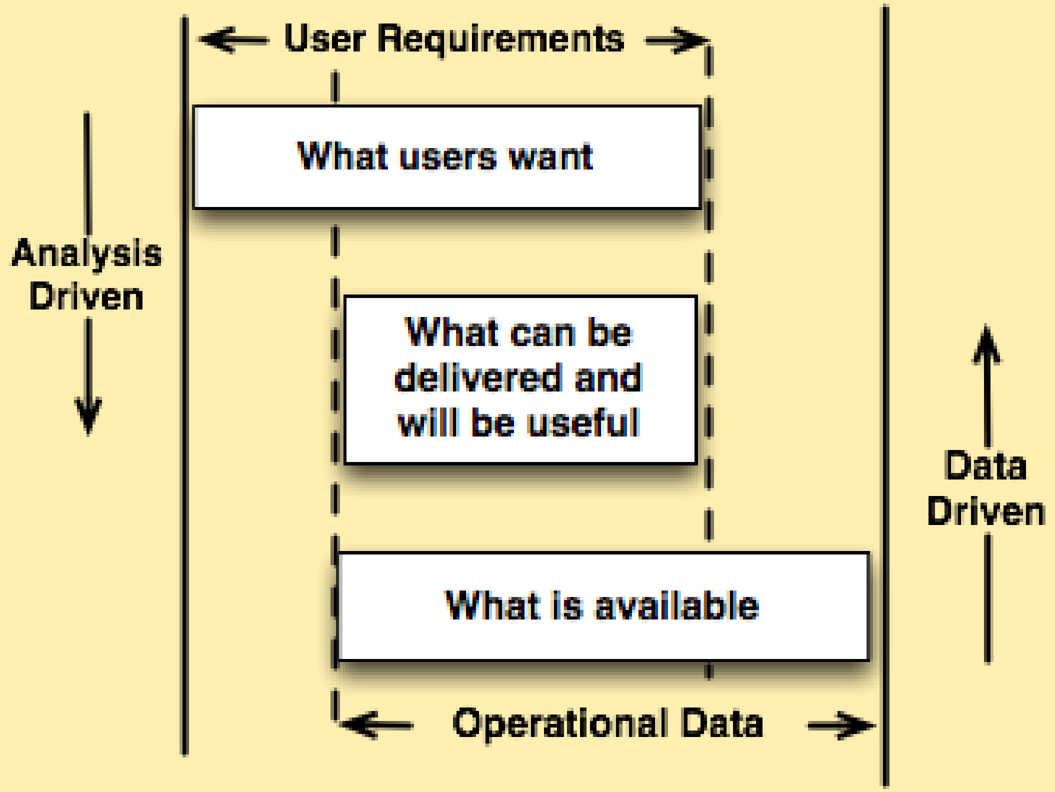
Data Driven (Top-Down, Data Push) Design - Inmon

An overall, big, enterprise-wide DW is designed, and then separate data marts are tailored for each business process. This approach would take longer to build the DW, and has a high cost and a high risk of failure.



DW CONCEPTUAL DESIGN APPROACHES

A combination of both methods



Think big, start small and avoid a costly 'big-bang' approach.

A DW DESIGN METHODOLOGY

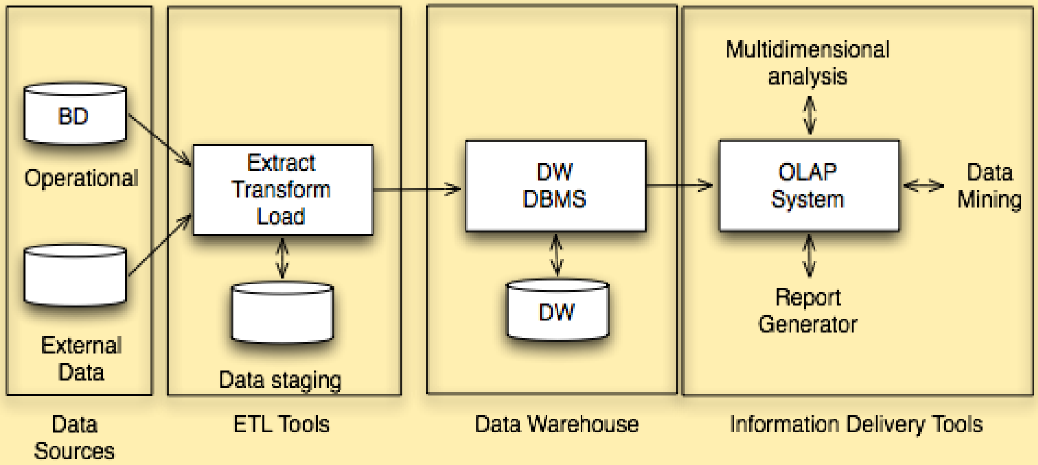
For each **data mart**:

- Requirements analysis
- Conceptual design
  - **Initial** conceptual design (what will be useful)
  - **Candidate** conceptual design (what can be delivered)
  - **Final** conceptual design (what can be analysed)
- Logical design

DW Logical design

DW Physical design

ETL (Extract, Transform and Load) Design



ANALYSIS DRIVEN DESIGN APPROACH

The choice of the first data mart to be implemented is of fundamental importance.

It should be the one that is most likely to be delivered on time, within budget, and to answer the most commercially important business questions.

The best choice for the first data mart tends to be the one that is related to **sales**.

REQUIREMENTS ANALYSIS

The purpose of a data warehouse **is not just to store data** but **rather to facilitate decision making.** As such, the first step to designing the schema for a data warehouse is to identify the different types of analyses that are relevant to business users.

Requirements gathering & Requirements specification

REQUIREMENTS GATHERING

- **Analysis of the nature and purpose of the business**
  - CelPhone is a company that deals with the production and sale of cellular phones with its own sales outlets
- **Interview business users who will provide information about the business processes and how they are measured**
  - Inventory process
  - Sales process
- **Interview data source system experts**

- Collection of business requirements for data analysis (business questions)

Process

N.	Business questions

Number of times a product has reached a minimum quantity

Inventory process

1	Average quantity on hand and reorder level for each model by month, by model identifier and description, by manufacturing plant, name and region.
2	Models that have reached the reorder level at least once in all manufacturing plants of a certain region.

REQUIREMENTS SPECIFICATION

Business process requirements

Process

N.	Business questions	Dimensions	Measures	Metrics

Inventory process

N	Business questions	Dimensions	Measures
1	Average quantity on hand and re-order level for each model by month, by model identifier and description, by manufacturing plant, name and region.	Model (ModelID Description), Manufactory (Name, Region), Date(Month)	QuantityOnHand, ReorderLevel
2	Models that have reached the re-order level at least once in all manufacturing plants of a certain region.	Model, Date(Week), Manufacturing(Region)	ReorderLevel

Metrics?

Fact description

Fact

Grain Description Fact Type	Preliminary Dimensions	Preliminary Measures

Inventory fact

Description	Preliminary Dimensions	Preliminary measures
A fact is about each product state at the end of the month.	Model, Manufactory, Date	QuantityOnHand, ReorderLevel

REQUIREMENTS SPECIFICATION

Dimensions

Dimensions

Name	Description	Granularity

Dimensions

Name	Description	Granularity
Date	...	A month
Model	...	A model
Manufactory	...	A manufacturing plant

Date

Attribute	Description
Month	...
Year	...

Dimensional attributes

Dimension

Attribute	Description

Model

Attribute	Description
ModelID	...
Description	...

Manufactory

Attribute	Description
Name	...
Region	...

Dimensional Hierarchies

Dimensional Hierarchies

Dimension	H. Description	H. Type

Dimensional Hierarchies

Dimension	Description	Hierarchy type
Date	Day $\rightarrow$ Month $\rightarrow$ Quarter $\rightarrow$ Year	Balanced

Dimensional attributes changes (see later on)

Dimension

Name	Changing attribute	Treatment of changes

MEASURES

Fact Measures and Metrics

Measure	Description	Aggregability	Calculated

Measures

Measure	Description	Aggregability	Calculated
QuantityOnHand	...	Semi additive across Date	No
ReorderLevel	...	Non additive	No

Descriptive attributes of the fact

Descriptive attributes

Attribute	Description

Measures

Measure	Description	Aggregability	Calculated
QuantityOrdered (Q)	...	Additive	No
ExtendedPrice (P)	UnitPrice $\times Q$	Additive	Yes
ExtendedCost (C)	UnitCost $\times Q$	Additive	Yes
Discount (D)	ExtendedPrice reduction	Additive	No
Revenue (R)	$P - D$	Additive	Yes
Margin	$R - C$	Additive	Yes

If there are several facts, two conformance matrix for common dimensions and measures

Facts Dimensions

Dimension	Fact 1	...	Fact n

Facts Measures

Measure	Fact	...	Fact n

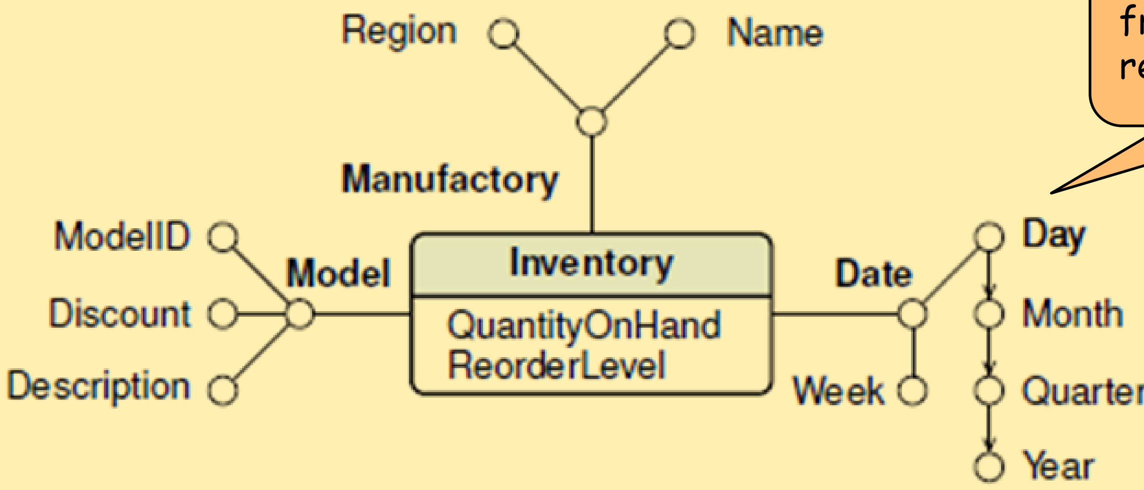
Fact dimensions

Dimension	Inventory	Sales
SalesOutlet		X
Model	X	X
Manufactory	X	X
Customer		X
Date	X	X

Fact measures

Measure	Inventory	Sales
QuantityOnHand	X	
ReorderLevel	X	
ExtendedPrice		X
ExtendedCost		X
Revenue		X
Margin		X
QuantityOrderd		X
Discount		X

INITIAL ANALYSIS-DRIVEN DATA MART CONCEPTUAL DESIGN



(a) *Inventory*

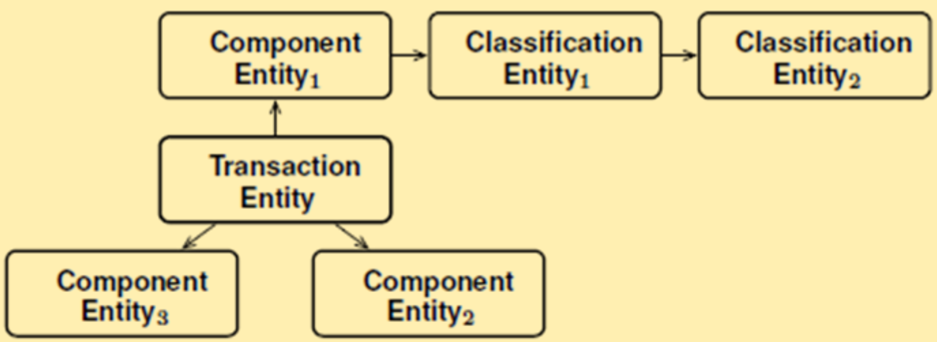
Day granularity comes from Sales data mart requirements

CANDIDATE DATA-DRIVEN CONCEPTUAL DESIGN

1. Operational databases analysis

2. Entity classification

   - **Transaction entities:** describe events that occur at a point in time and contain measurements
   - **Component entities:** related to transaction entities via a one-to-many relationship
   - **Classification entities:** related to component entities via a one-to-many relationship chain



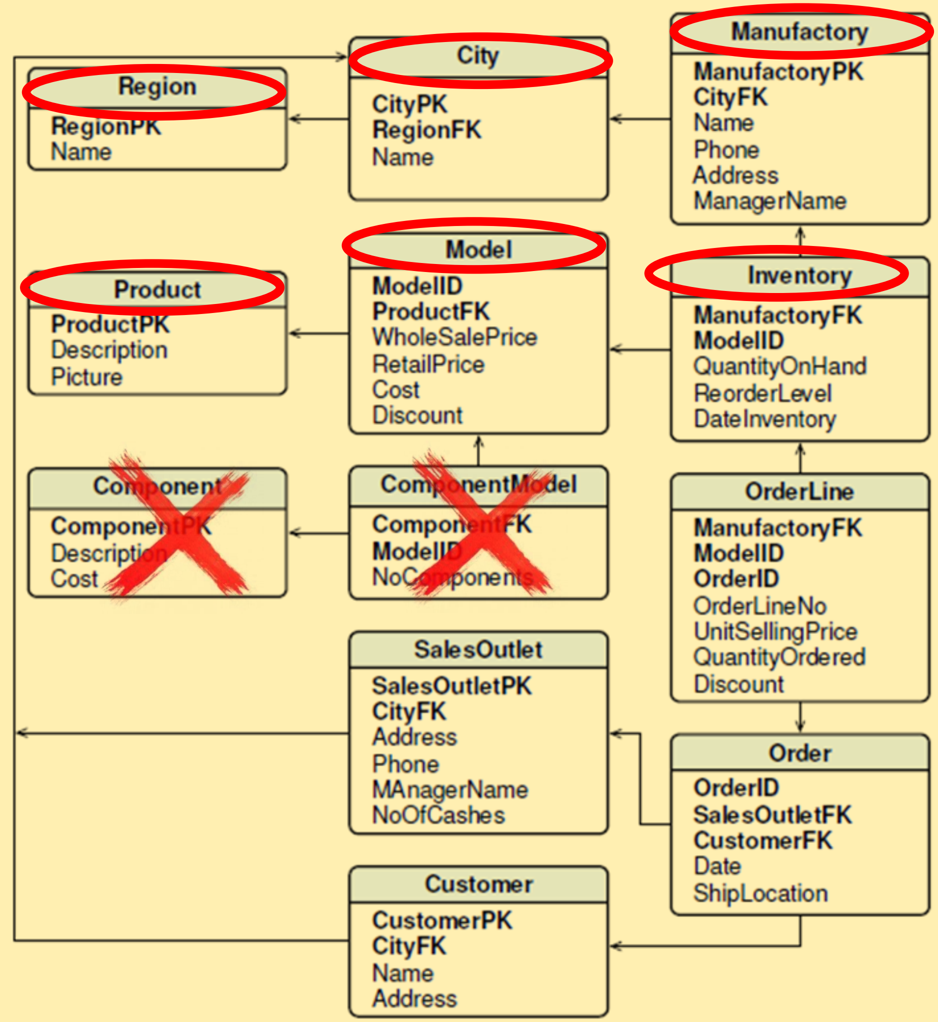
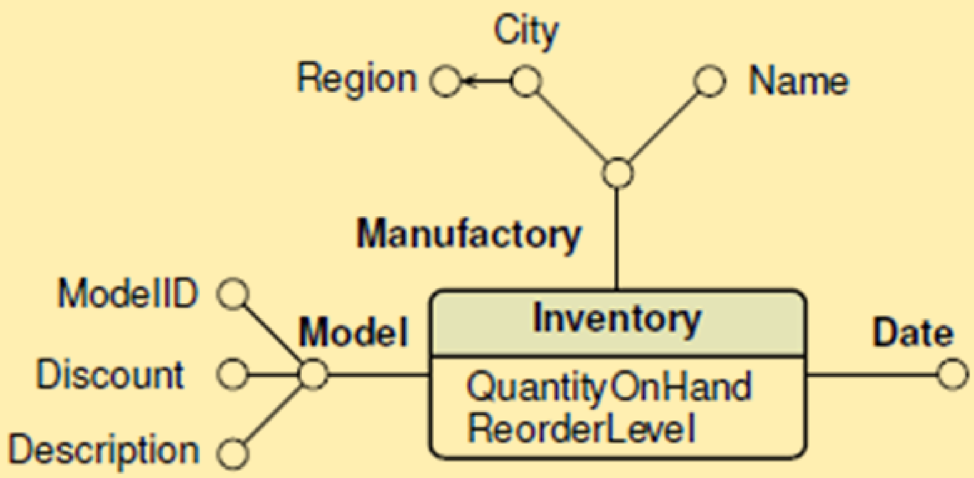


Figure 3.13: The operational database

CANDIDATE DATA-DRIVEN CONCEPTUAL DESIGN

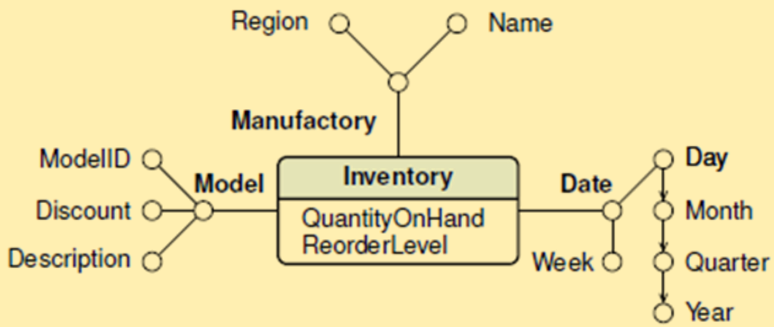
3. Candidate data mart conceptual design



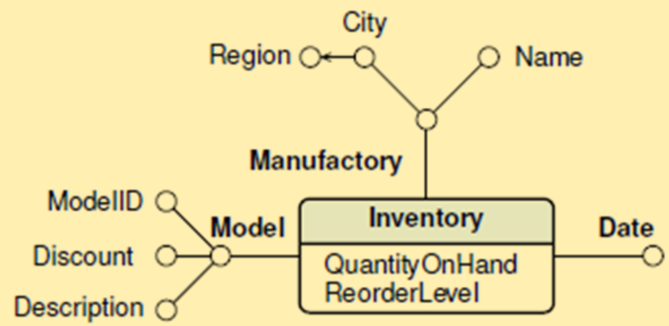
(a) *Inventory*

FINAL DATA MART CONCEPTUAL DESIGN

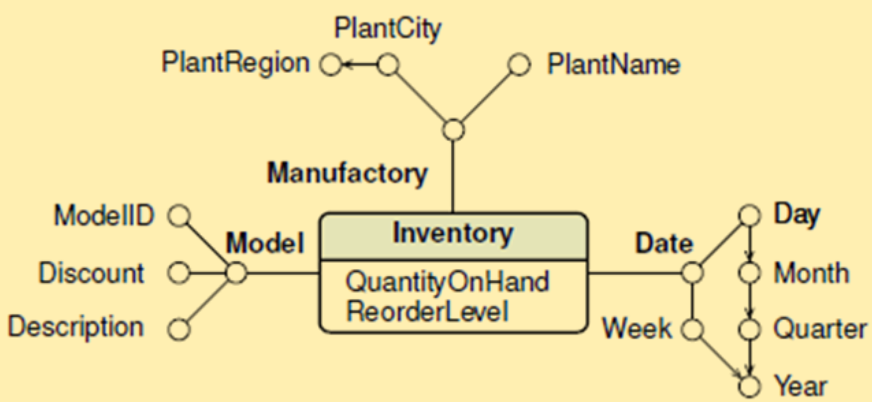
- From a comparison of the initial and candidate conceptual designs the final data marts are defined (the design of *will be useful* and *what can be delivered*)



(a) *Inventory*



(a) *Inventory*



(a) *Inventory*

FROM CONCEPTUAL TO LOGICAL DESIGN

DB **DW**

CONCEPTUAL DESIGN

Object Data Model

Dimensional fact model

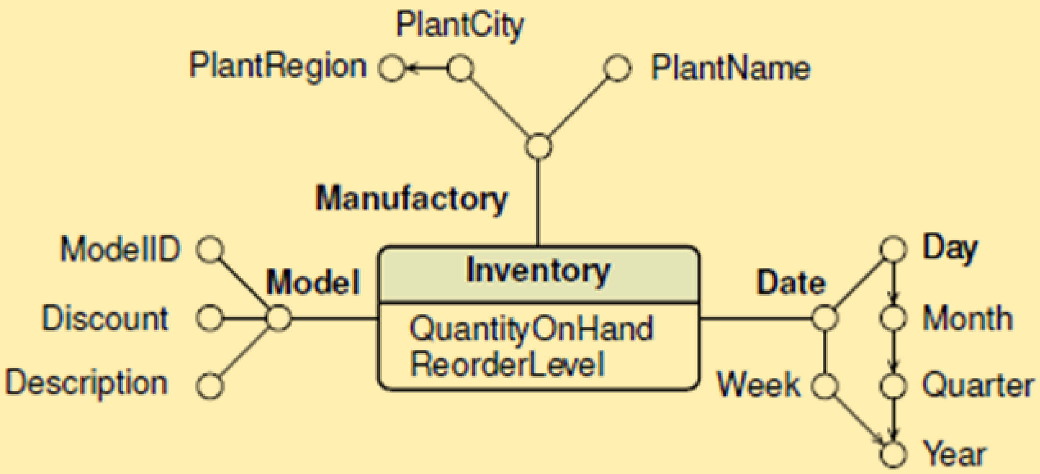
LOGICAL DESIGN

Selection of a DBMS and definition of a **normalized** relational schema

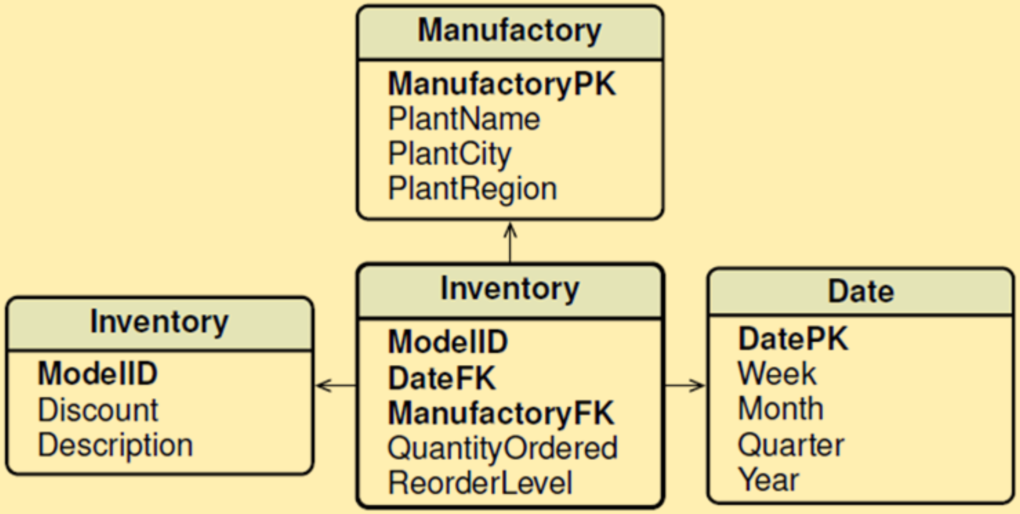
Selection of a DWMS and definition of a relational data mart, to **enhance performance** and maintaining **an intuitive user interface**.

Usually **normalizing the dimension tables** would interfere with both of these objectives.

LOGICAL DESIGN: THE SIMPLE CASE

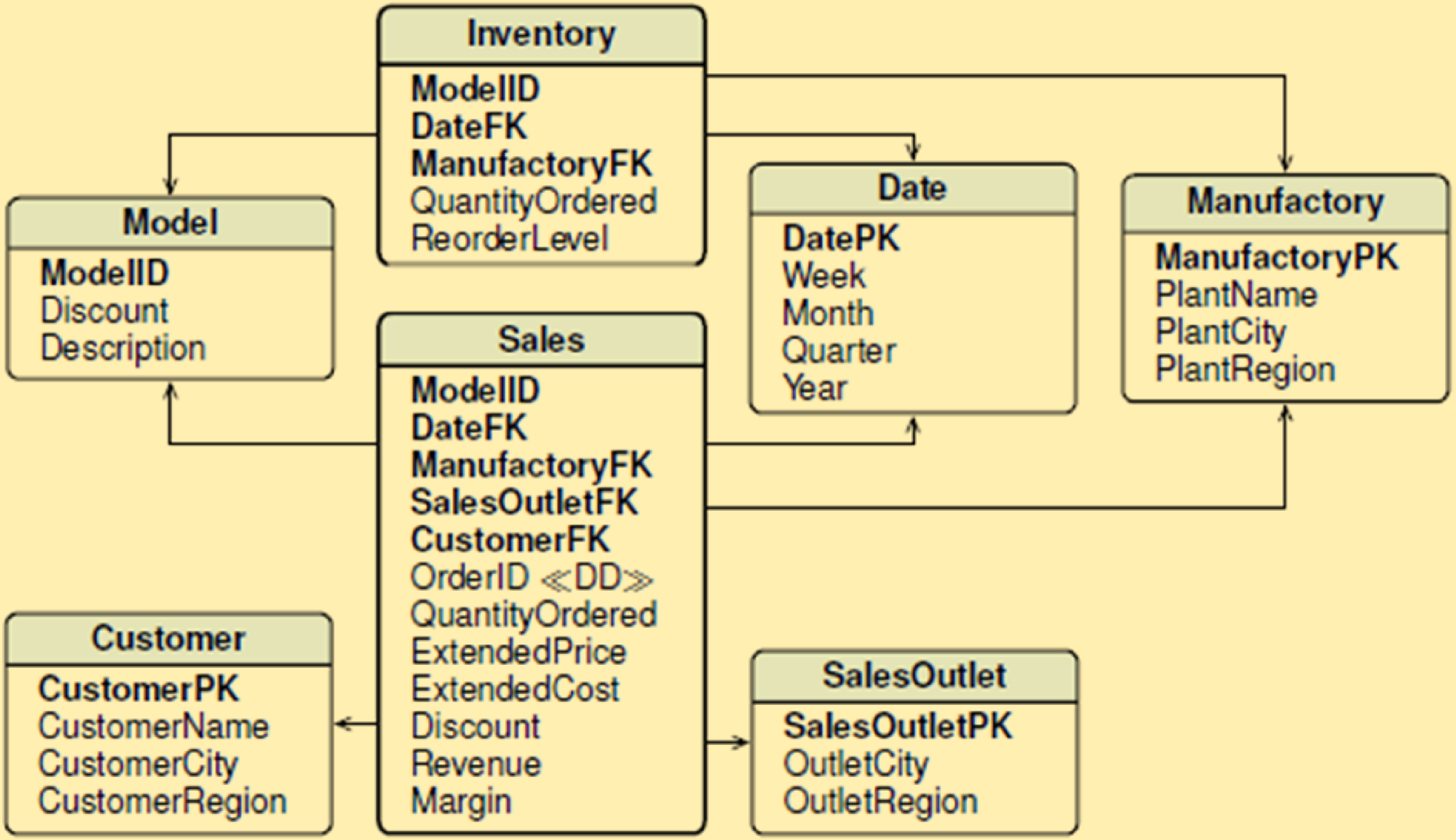


(a) *Inventory*



(a) *Inventory data mart star schema*

LOGICAL DESIGN: THE SIMPLE CASE



(c) *Data warehouse constellation schema*

Date(DatePk, Month, Quarter, Year)

How to verify on the loaded table the validity of the hierarchy **Month → Year ?**

Write a query that returns an empty result set
iff **the functional dependency is valid.**

■ **Definition 8.1** *Functional Dependency*

Given a relation schema R and X, Y subsets of attributes of R, a functional dependency $X \to Y$ (X *determines* Y) is a constraint that specifies that for every possible instance r of R and for any two tuples $t_1, t_2 \in r$, $t_1[X] = t_2[X]$ implies $t_1[Y] = t_2[Y]$.

EXERCISE: TEST DIMENSIONAL HIERARCHIES … … AND YOUR KNOWLEDGE OF SQL

Date(DatePk, Month, Quarter, Year)

How to verify on the loaded table the validity of the hierarchy **Month** → **Year ?**

Write a query that returns an empty result set
iff **the functional dependency is valid.**

```
SELECT    Month
FROM      Date
GROUP BY  Month
HAVING    COUNT(DISTINCT Year) > 1;
```

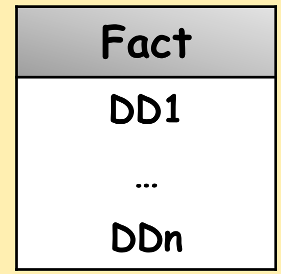
```
WITH MonthYearSubquery AS
    (SELECT DISTINCT  Month, Year
     FROM       Date)
SELECT    Month
FROM      MonthYearSubquery
GROUP BY  Month
HAVING    COUNT(*) > 1;
```

DEGENERATE DIMENSIONS

Always stored in the fact table?

Space to store in the fact table is

[space(DD1) + … + space(DDn)]*NFacts



A junk dimension contains all possible combinations

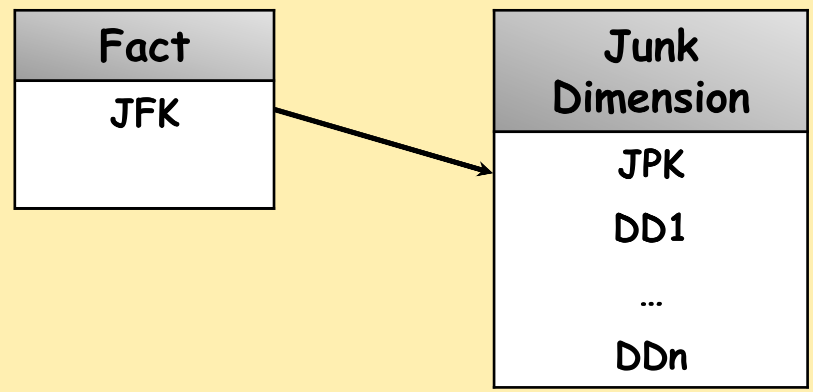
of values of DD1, …, DDn

Space with a junk dimension is

space(JFK)*Nfacts +

[space(JPK)+space(DD1) + … + space(DDn)]

* NValues1 * … * NValuesn



Which solution is more convenient?

DEGENERATE DIMENSIONS: EXAMPLE

- DD1 is age, space(DD1) = 8 bytes, Nvalues1 = 100
- DD2 is gender, space(DD2) = 1 byte, NValues2 = 2
- space(JFK) = space(JPK) = 8 bytes
- Space to store in the fact table: 9 x NFacts
- Space to store in junk dimension: 8 x NFacts + (8+ 8 + 1) x 100 x 2
- Junk dimension is better for 8 NFacts + 3400 < 9 NFacts, i.e., 3400 < NFacts

Fact

…	Age	Gender
…	18	M
…	100	F
…	…	…

Fact

…	JFK
…	1
…	164
…	…

Junk Dimension

JPK	Age	Gender
1	18	M
2	18	F
…	…	…
163	100	M
164	100	F
…	…	…