

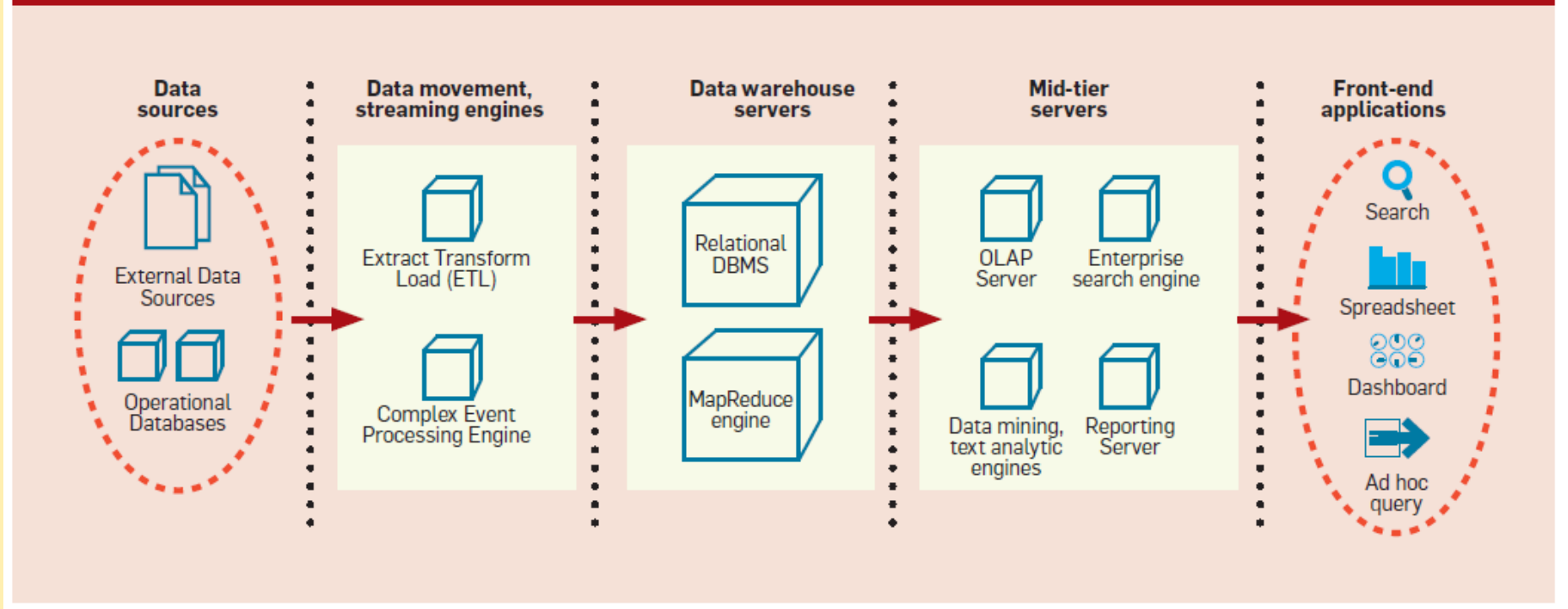
# BUSINESS INTELLIGENCE

**OLAP: On-Line Analytical Processing**

# BI Architecture

2

**Figure 1. Typical business intelligence architecture.**



# ON-LINE ANALYTICAL PROCESSING (OLAP)

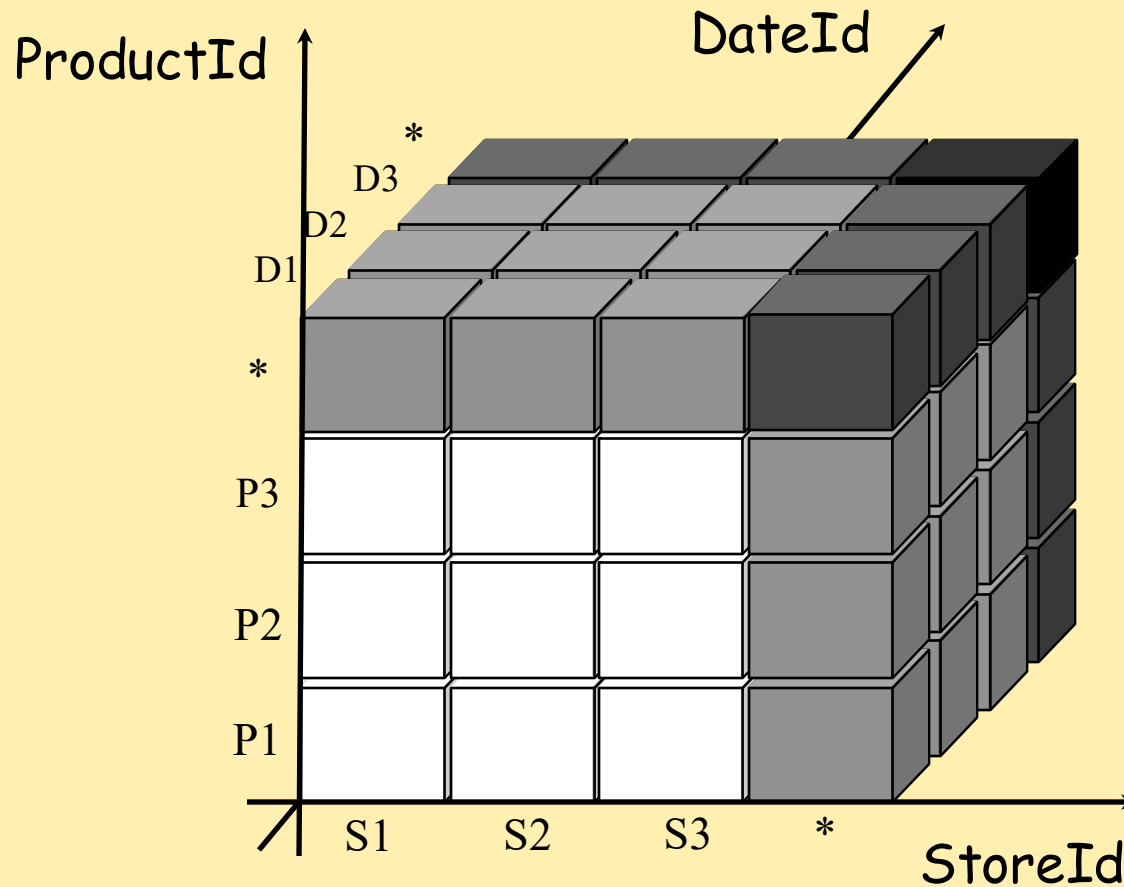
- An OLAP server provides a *multidimensional view* starting from a datawarehouse



- The multidimensional view can be navigated through pivot tables, reports, 2-D or 3-D plots, or it can be queried using a query language (eg., MDX - MultiDimensional eXpressions)

# MULTIDIMENSIONAL MODEL (CUBE)

The multidimensional model is useful to understand interactive data analysis, and how to improve the execution performance.



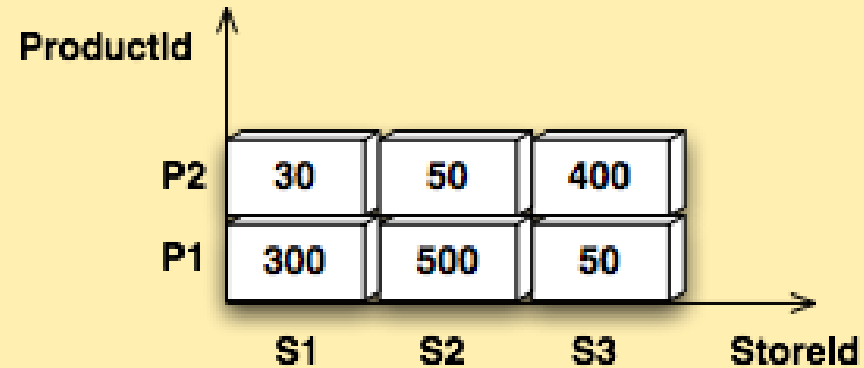
## 2-D CUBE

M

Sales

StoreId	ProductId	Qty
S1	P1	300
S2	P1	500
S3	P1	50
S1	P2	30
S2	P2	50
S3	P2	400

Fact Table



2-D Cube

## CROSS TABULATION

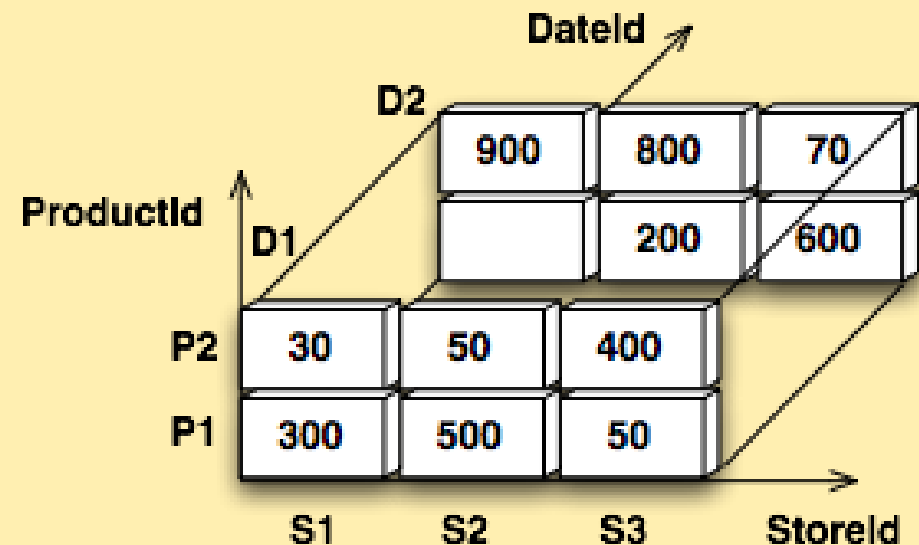
ProductId	StoreId		
	S1	S2	S3
P1	300	500	50
P2	30	50	400

# 3-D CUBE

Sales

StoreId	ProductId	DateId	Qty
S1	P1	D1	300
S2	P1	D1	500
S3	P1	D1	50
S1	P2	D1	30
S2	P2	D1	50
S3	P2	D1	400
S2	P1	D2	200
S3	P1	D2	600
S1	P2	D2	900
S2	P2	D2	800
S3	P2	D2	70

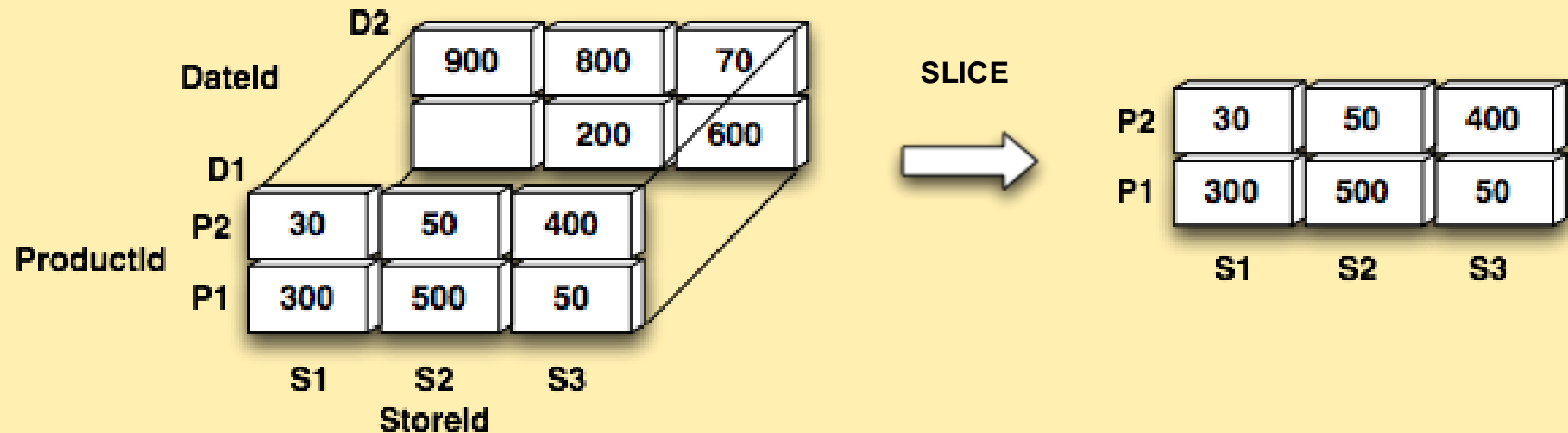
Fact Table



3-D Cube

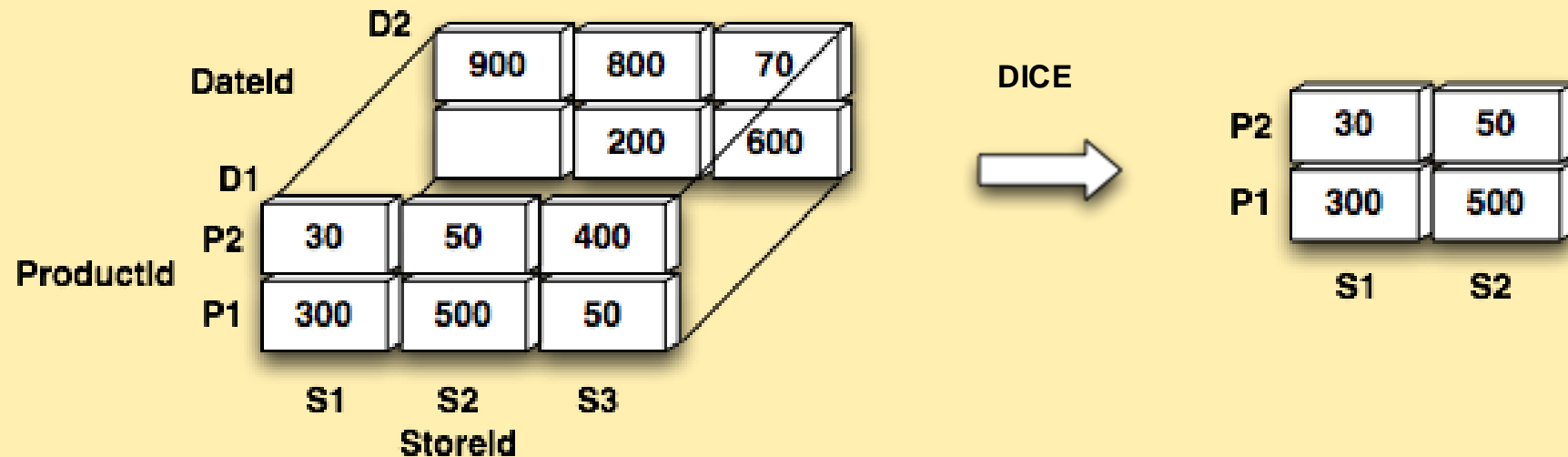
## CUBE OPERATOR: SLICE

Sales **SLICE** FOR DateId = 'D1';



# CUBE OPERATOR: DICE

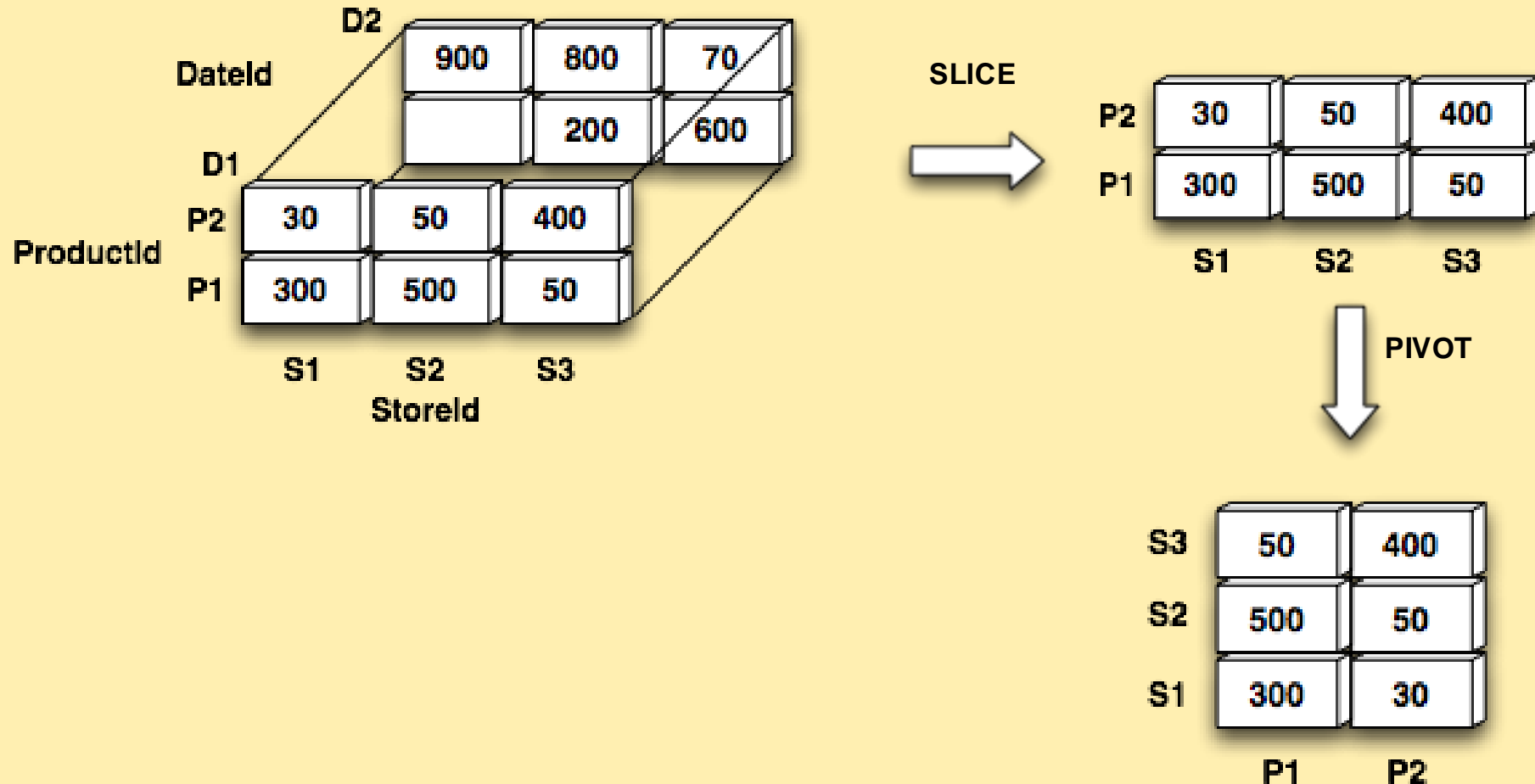
Sales **DICE FOR** DateId = 'D1'  
StoreId **IN** ('S1', 'S2');





# CUBE OPERATOR: PIVOT

**PIVOT** (Sales *SLICE* FOR DateId = 'D1');



**Rotate:** reorient the cube, visualization, 3D to series of 2D planes

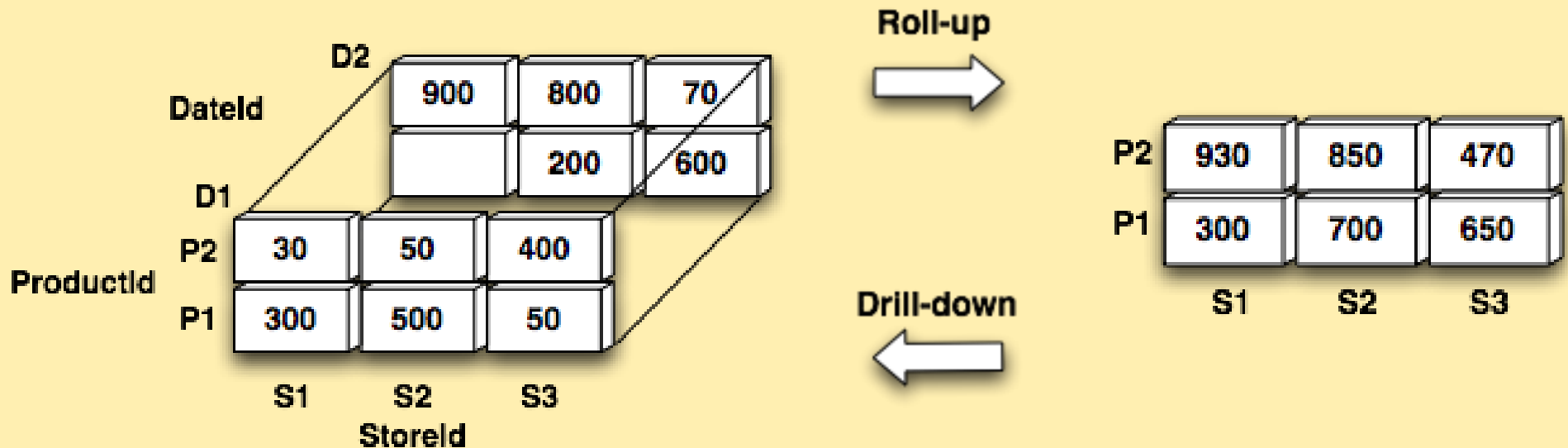
# CUBE OPERATORS: ROLL-UP and DRILL-DOWN

Roll-up aggregates data by **dimension reduction** or **by navigating attribute hierarchy** (Drill-down is the reverse of roll-up)

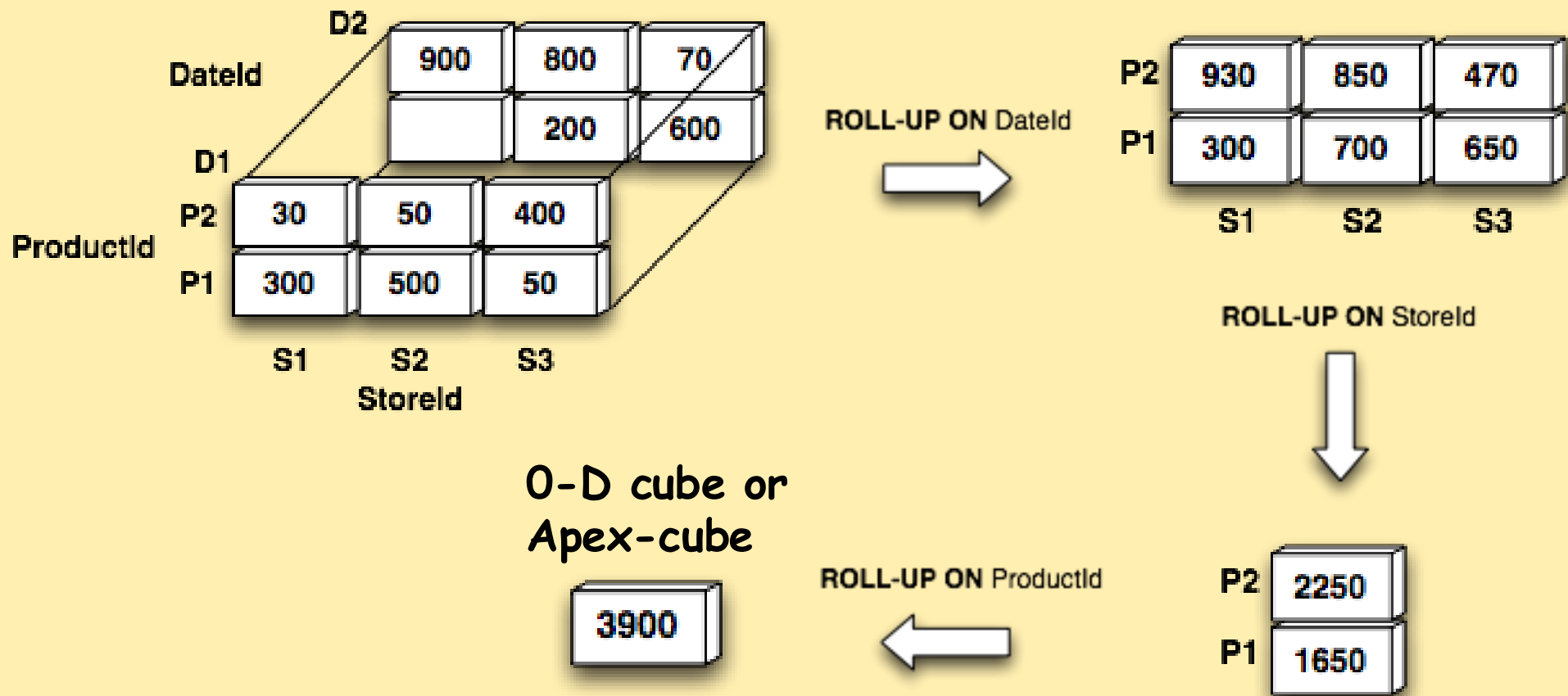
Hypothesis: one measure and aggregations by **sum**.

## SALES ROLL-UP ON DateId

(total Qty by ProductId and by StoreId)



# CUBE OPERATORS: ROLL-UP and DRILL-DOWN



# CUBE OPERATORS: DRILL THROUGH

Drill-through produces the facts that satisfy a cell coordinate

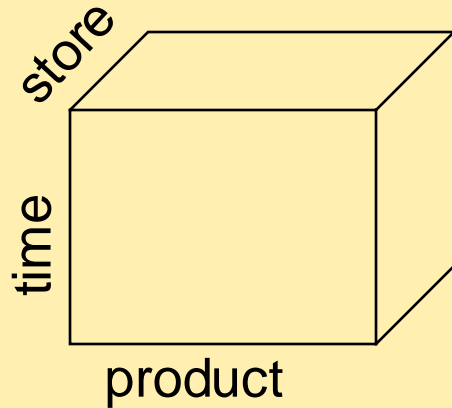
Sales

StoreId	ProductId	DateId	Qty
S1	P1	D1	300
S2	P1	D1	500
S3	P1	D1	50
S1	P2	D1	30
S2	P2	D1	50
S3	P2	D1	400
S2	P1	D2	200
S3	P1	D2	600
S1	P2	D2	900
S2	P2	D2	800
S3	P2	D2	70

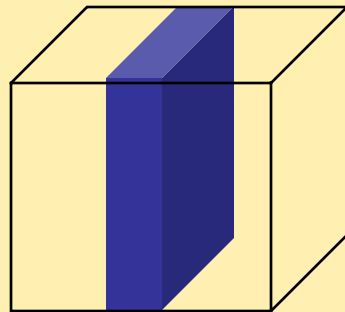
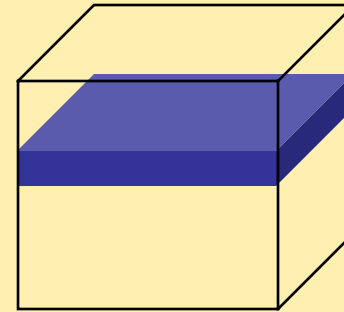
P2	2250
P1	1650

StoreId	ProductId	DateId	Qty
S1	P2	D1	30
S2	P2	D1	50
S3	P2	D1	400
S1	P2	D2	900
S2	P2	D2	800
S3	P2	D2	70

# CUBE NAVIGATION BY DIFFERENT USERS

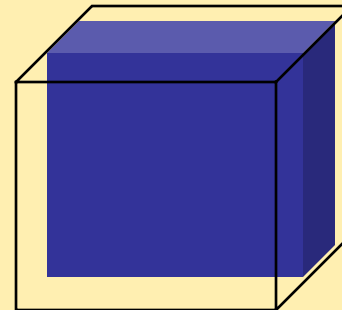


Finance manager look at sales  
of a period compared to the previous period  
for any product and any market



Product managers look at sales  
of some products  
in any period and in any market

Branch manager look at sales  
of his/her stores  
for any product and any period



# TEXTUAL NOTATION FOR CUBE OPERATORS

**Hypothesis:** one measure and aggregations by **sum**.

**Sales(StoreId, ProductId, DateId)**

is the cube with dimensions **StoreId**, **ProdottoId**, **DataId**, and measure **M**

A cube operation is denoted by substituting a dimension with a value

## TEXTUAL NOTATION FOR CUBE OPERATORS (cont)

Sales(StoreId, ProductId, 'D1')      **slice**

Sales('S1', ProductId, 'D1')      **dice**

Sales('S1', 'P1', 'D1')      **dice**

## TEXTUAL NOTATION FOR CUBE OPERATORS (cont.)

Each dimension domain is extended with the value “\*”, that means summarize data (**sum**) by all the dimension values.

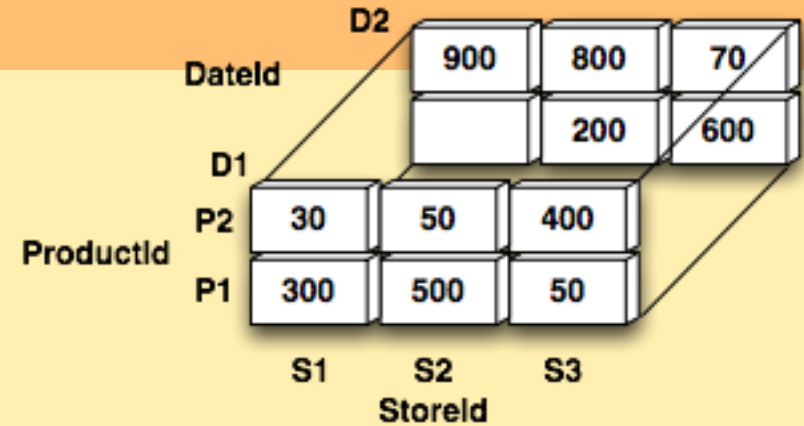
Sales(StoreId, ProductId, \*)

Sales by **roll-up** on DateId with **sum(M)**

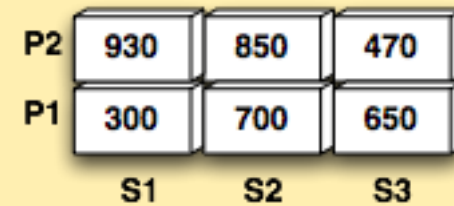


# CUBE OPERATORS: EXAMPLES

Sales(StoreId, ProductId, DateId) =



Sales(StoreId, ProductId, \*) =



Sales(StoreId, \*, \*) =



Sales(\*, \*, \*) =



# CUBE OPERATORS: EXAMPLES

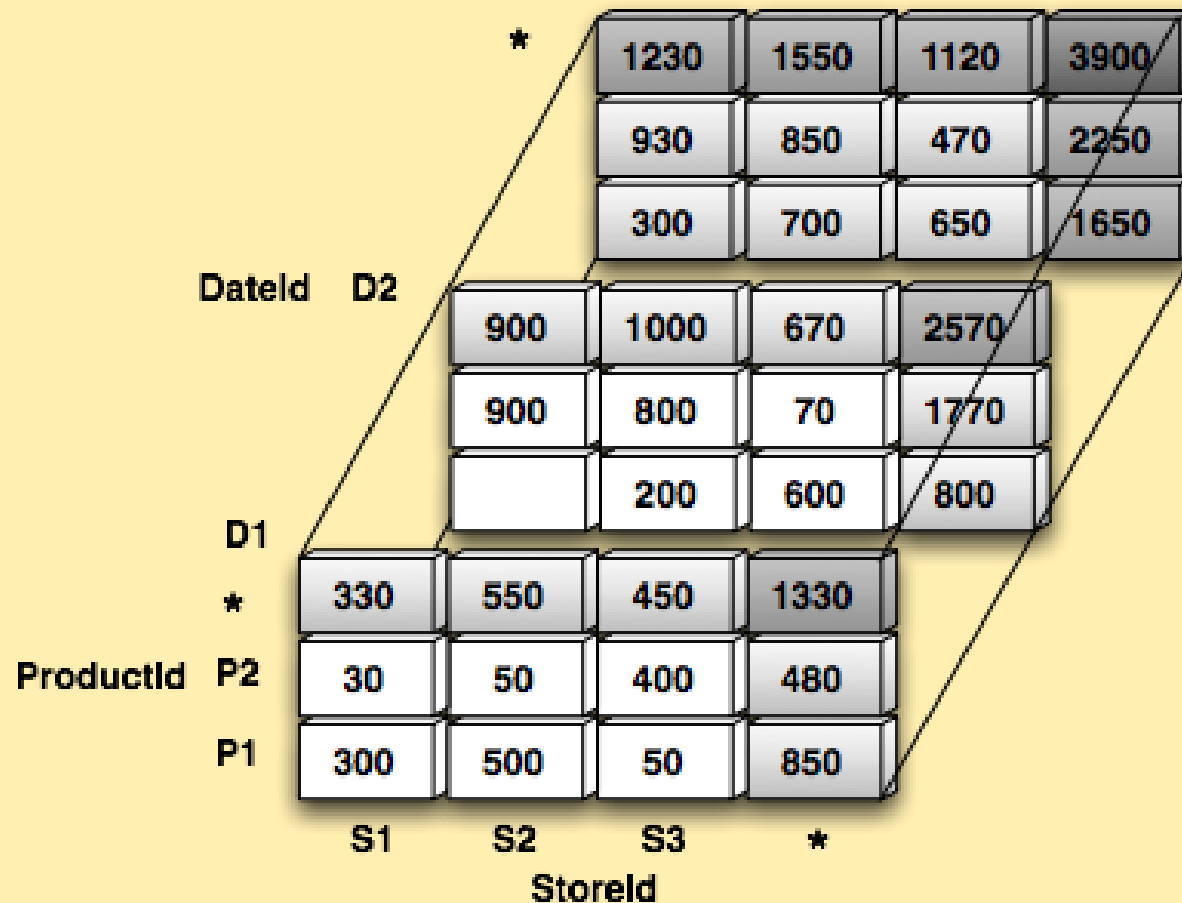
- What is

Sales(StoreId, 'P1', \*) =

P1	300	700	650
	S1	S2	S3

# EXTENDED CUBE

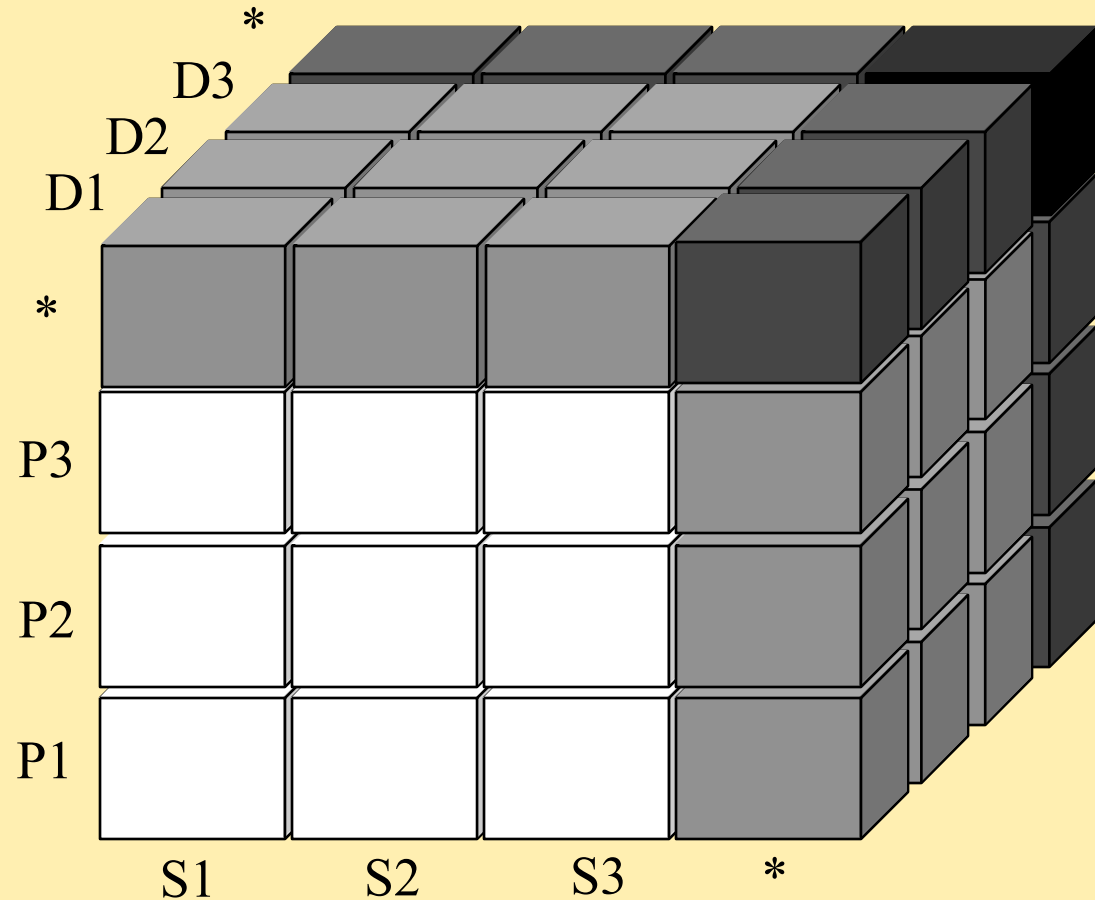
A data cube is extended with the value '\*' for each dimensions, and in the corresponding cells is stored the **sum of the measure**.



# EXTENDED CUBE

With the '\*' values, the cube becomes a set of **cuboids**:

- **white** cells are the data cube
- **gray** cells are roll-up by a dimension,
- **dark gray** cells are roll-up by two dimensions
- **black** cells are roll-up by all dimensions.



# EXTENDED CROSS TABULATION

Sales

StoreId	ProductId	Qty
S1	P1	300
S2	P1	500
S3	P1	50
S1	P2	30
S2	P2	50
S3	P2	400

## CROSS TABULATION

ProductId	StoreId		
	S1	S2	S3
P1	300	500	50
P2	30	50	400

## EXTENDED CROSS TABULATION

ProductId	StoreId			Total
	S1	S2	S3	
P1	300	500	50	850
P2	30	50	400	480
Total	330	550	450	1330

**OLAP** refers to the technique of performing complex business analysis over the information stored in a data warehouse.

We will see how report developers use SQL to write queries, but there are business intelligence tools that allows a user or a developer to make data analysis and to build beautiful reports without any knowledge of SQL...  
which is generated automatically.