# DECISION SUPPORT SYSTEMS Module I: DECISION SUPPORT DATABASES

**Course home page,teaching material, and recorded lessons:**

http://didawiki.di.unipi.it/doku.php/mds/dsd/

- Today, an introduction to
  - **What** is a Data Warehouse (DW)
  - **What** do we model in a DW
- and some tests of your knowledge of SQL
  - we will recall **relational algebra** and **SQL** starting next week

# DEFINITION



*A* **DW** is a **decision support database** with **historical**, **nonvolatile data**, pulled together primarily from operational business systems, structured and tuned **to facilitate analysis of the performance of key business processes**, **worthy of improvement**.

The first definition of data warehouse was provided by William Inmon in 1990, One of the two fathers of DW (the other is Ralph Kimball)

*A* **DW** is a **specialized database**

- **static (non volatile)**,
- **with integrated data** from different data sources,
- **organized to analyze subjects of interest**,
- **with historical data**,
- **used to produce summarized data to support decision-making processes**.

# WHY SEPARATE DB AN DW?



**To promote the high performance of both systems**

- Special data organization, and implementation techniques are needed to support multidimensional OLAP analysis.
- Complex data analysis would degrade performance of operational DBMS.

**The systems have different structures, contents, and uses of the data**

- Decision support requires historical data which operational DBs do not typically maintain.
- Decision support requires aggregation of data from heterogeneous sources: operational DBs, external sources.
- Different sources typically use inconsistent data representations, codes and formats which have to be reconciled.

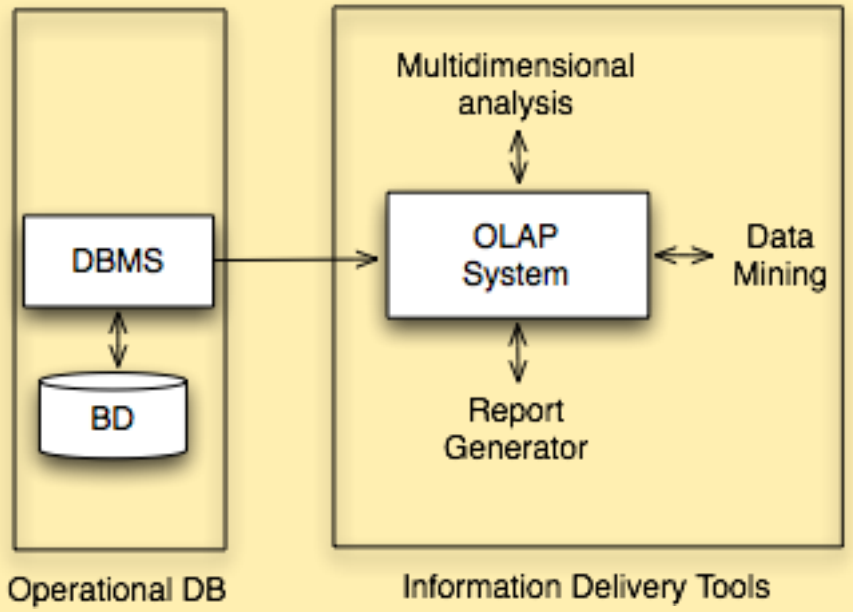
**Data warehousing** is the process to bring data from operational (OLTP) sources into a single data warehouse for (OLAP) analysis with Business Intelligence applications.

**OLTP (On Line Transaction Processing)**

**OLAP (On Line Analytical Processing)**

## Single-Layer Architecture



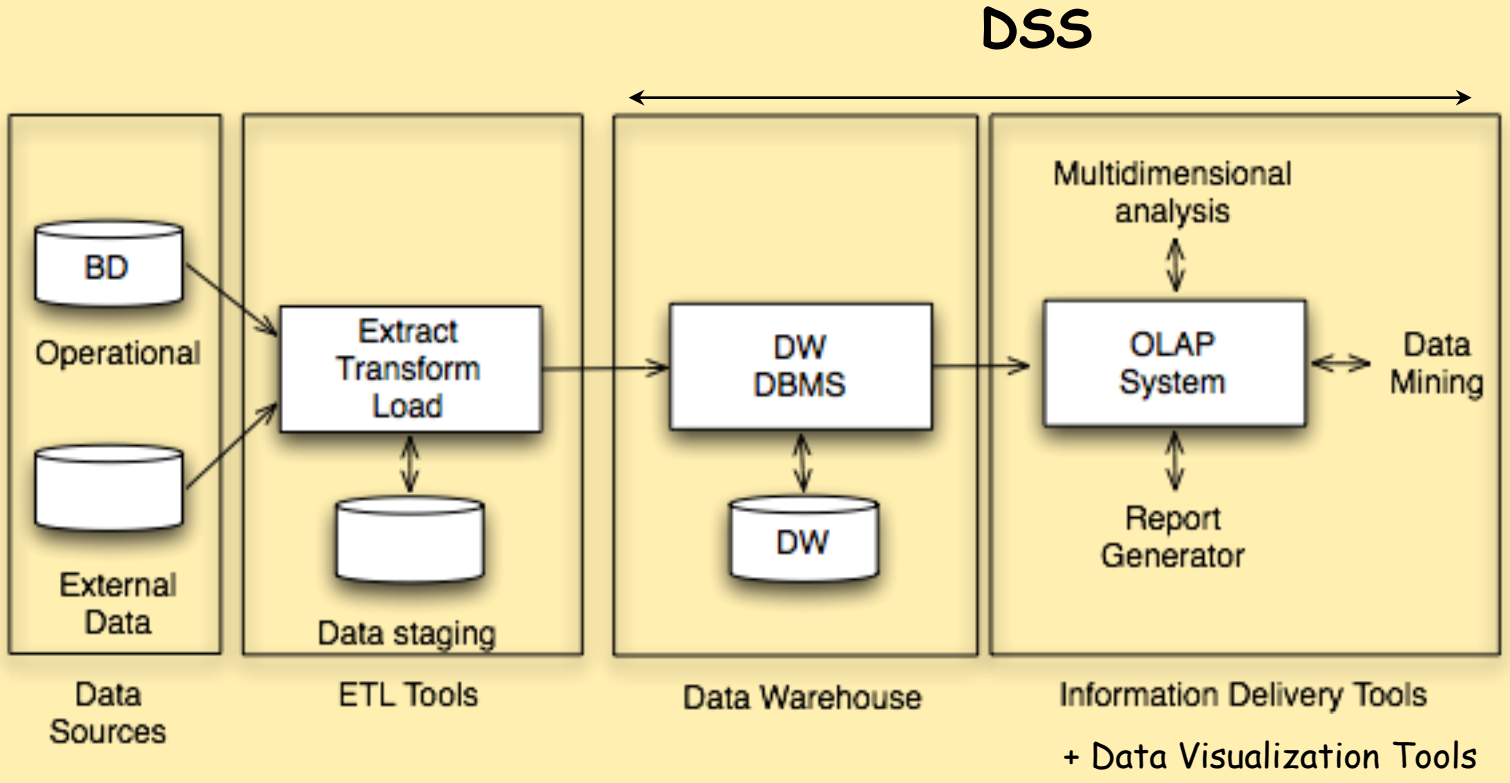



**Business Intelligence**

**OLAP (On Line Analytical Processing)**

## Three-Layer Architecture






OLAP (On Line Analytical Processing)

# WHAT IS MODELED IN A DW?



**Computer science** is a **science of abstraction**:

Creating the right **model** for thinking about a real-world problem that can be understood by computer users and that can be represented and manipulated inside a computer.

The attention here is on modeling the world in data organized as **collections of logically related structured data** called **decision support databases (DW)**.

**What is modeled in a DW?**

# EXAMPLE:

https://community.microstrategy.com/s/gallery




| LQ Revenue | LQ Profit | Category | Revenue | Profit | Sell-through % | Order Count | Units Received | End on hand |
|---|---|---|---|---|---|---|---|---|
| $12,216 | $911 | Music | $13,039 | $737 | 5.9% | 886 | 27,865 | 14,064 |
| $60,168 | $4,644 | Movies | $64,021 | $3,376 | 27.2% | 4,135 | 21,685 | 11,075 |
| $268,185 | $49,368 | Electronics | $285,029 | $47,573 | 24.8% | 832 | 4,810 | 2,526 |
| $12,846 | $2,972 | Books | $13,023 | $2,676 | 48.3% | 850 | 495 | 911 |

# WHAT IS MODELED IN A DW?



**Managers think about a business process in terms of**

| | |
|---|---|
| **facts,** | A **fact** is an observation of the performance of a business process (the subject of analysis) (e.g. a sale) |
| **measures,** | The **measures** are numerical attributes of a fact (e.g. qty, revenue, etc), |
| **dimensions,** | The **dimensions** give facts their **context.** In general a dimension is described by a **set of attributes**, otherwise is called **degenerate**. (e.g. sales revenue by **product** category, by month **time**, and by city **market**). |
| and **hierarchies,** | The **attributes** of a dimension may be related via a **hierarchy** of relationships (e.g. a month is related to the quarter and the year attributes). |

**Managers are interested in aggregate data:** the sum, average minimum, maximum, ..., of measures of data groups with equal values of some dimensions or dimensional attributes.

**Metrics** and **Key Performance Indicators** (KPI)

**Total sales revenue, by products.**

## Total revenue, by Product (SQL ?)

**SALES**

| Product | Store | Date | Revenue |
|---|---|---|---|
| p1 | m1 | d1 | 120 |
| p2 | m1 | d1 | 110 |
| p1 | m3 | d1 | 500 |
| p2 | m2 | d1 | 800 |
| p1 | m1 | d2 | 400 |
| p1 | m2 | d2 | 300 |

| Product | Store | Date | Revenue |
|---|---|---|---|
| p1 | m1 | d1 | 120 |
| p1 | m3 | d1 | 500 |
| p1 | m1 | d2 | 400 |
| p1 | m2 | d2 | 300 |
| p2 | m1 | d1 | 110 |
| p2 | m2 | d1 | 800 |

| Product | Total Revenue |
|---|---|
| p1 | 1320 |
| p2 | 910 |

## Total revenue, by Product (SQL ?)

**SALES**

| Product | Store | Date | Revenue |
|---|---|---|---|
| p1 | m1 | d1 | 120 |
| p2 | m1 | d1 | 110 |
| p1 | m3 | d1 | 500 |
| p2 | m2 | d1 | 800 |
| p1 | m1 | d2 | 400 |
| p1 | m2 | d2 | 300 |

```
SELECT   Product
         , SUM(Revenue) AS TotalRevenue
FROM     Sales
GROUP BY Product ;
```

| Product | Store | Date | Revenue |
|---|---|---|---|
| p1 | m1 | d1 | 120 |
| p1 | m3 | d1 | 500 |
| p1 | m1 | d2 | 400 |
| p1 | m2 | d2 | 300 |
| p2 | m1 | d1 | 110 |
| p2 | m2 | d1 | 800 |

| Product | Total Revenue |
|---|---|
| p1 | 1320 |
| p2 | 910 |

**Managers think in term of business dimensions** to analyze the data.

**Total revenue** by Product.
**Total revenue** by Product, by Market

| Revenue by Product | |
|---|---|
| **Product** | **Revenue (€)** |
| P1 | 130 |
| P2 | 910 |

| Revenue by Product and Market | | |
|---|---|---|
| **Product** | **Market** | **Revenue (€)** |
| P1 | M1 | 520 |
| | M2 | 300 |
| | M3 | 500 |
| P2 | M1 | 110 |
| | M2 | 800 |

# FACTS ANALYSIS WITH SUBTOTALS



**Revenue by Product and Market**

| Product | Market | Revenue (€) |
|---|---|---|
| P1 | M1 | 520 |
| | M2 | 300 |
| | M3 | 500 |
| **P1** | **Total** | **1 320** |
| P2 | M1 | 110 |
| | M2 | 800 |
| **P2** | **Total** | **910** |
| **Total** | | **2 230** |

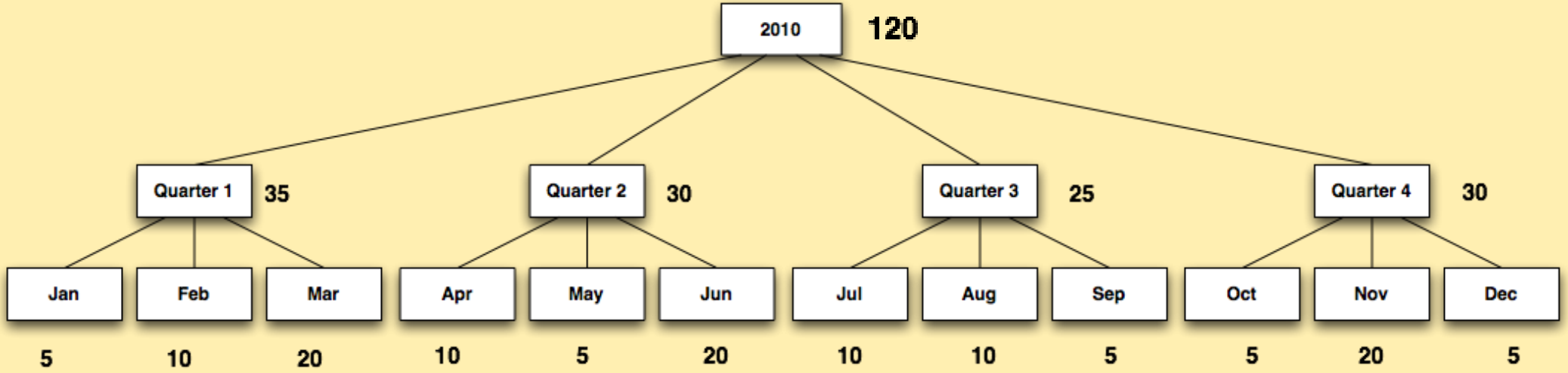
Managers analyse measure aggregates by business dimensions, and then in various levels of details, by exploiting dimensional attributes hierarchies.

**Total Revenue, by Month**

**Total Revenue, by Quarter**

**Total Revenue, by Year**

**Example with Sales of Year 2010**



A dimensional attributes hierarchy models **attributes dependency**, i.e. a **functional dependency** between attributes, using the relational model terminology.

**Definition 8.1** *Functional Dependency*

Given a relation schema $R$ and $X, Y$ subsets of attributes of $R$, a functional dependency $X \rightarrow Y$ ($X$ *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 $t_1[Y] = t_2[Y]$.

For example, the dimension **Date** has attributes **Month**, **Quarter**, **Year**. Can we define a **dimensional hierarchy** among them?

**Month → Quarter → Year**

**Date** **Month → Quarter → Year**

| PkDate | Month | Quarter | Year |
|---|---|---|---|
| 20080101 | 1 | 1 | 2008 |
| 20080102 | 1 | 1 | 2008 |
| ... | | | |
| 20090101 | 1 | 1 | 2009 |
| 20090102 | 1 | 1 | 2009 |

**Date** **Month → Quarter → Year**

| PkDate | Month | Quarter | Year |
|---|---|---|---|
| 20080101 | 1 | 1 | 2008 |
| 20080102 | 1 | 1 | 2008 |
| ... | | | |
| 20090101 | 1 | 1 | 2009 |
| 20090102 | 1 | 1 | 2009 |

# USEFULNESS OF HIERARCHIES (SQL)



**Sales(Product,..., Month, Quarter, Year, Revenue)**

**Month → Quarter**

```
V = SELECT   Month, SUM(Revenue) AS SRM
    FROM     Sales
    WHERE    Year = 2010
    GROUP BY Month;
```

**For the year 2010, the total Revenue, by Month**

```
Q = SELECT   Quarter, SUM(Revenue) AS SRQ
    FROM     Sales
    WHERE    Year = 2010
    GROUP BY Quarter;
```

**For the year 2010, the total Revenue, by Quarter.**

**If the result of V is materialized, then may we compute Q from V only ?**

# USEFULNESS OF HIERARCHIES (SQL)



**Sales(Product,..., Month, Quarter, Year, Revenue)**

**Month → Quarter**

```
V' = SELECT   Month, Quarter, SUM(Revenue) AS SRMQ
     FROM     Sales
     WHERE    Year = 2010
     GROUP BY Month, Quarter;
```

**For the year 2010, the total Revenue, by Month, by Quarter**

```
Q = SELECT   Quarter, SUM(Revenue) AS SRQ
    FROM     Sales
    WHERE    Year = 2010
    GROUP BY Quarter;
```

**For the year 2010, the total Revenue, by Quarter.**

**If the result of V' is materialized, then may we compute Q from V' only ?**

```
Q = SELECT   Quarter, Sum(SRMQ) AS SRQ
    FROM     V'
    GROUP BY Quarter
```

# USEFULNESS OF HIERARCHIES AND DISTRIBUTIVE AGGREGATION FUNCTIONS



$$V = V_1 \cup V_2 \qquad V_1 \cap V_2 = \varnothing$$

**Distributive**

E.g., sum(), min(), max(), count()

**Algebraic**

E.g., avg(), standard_deviation().

**Holistic**

E.g., median(), mode(), rank().