
Lecture 03

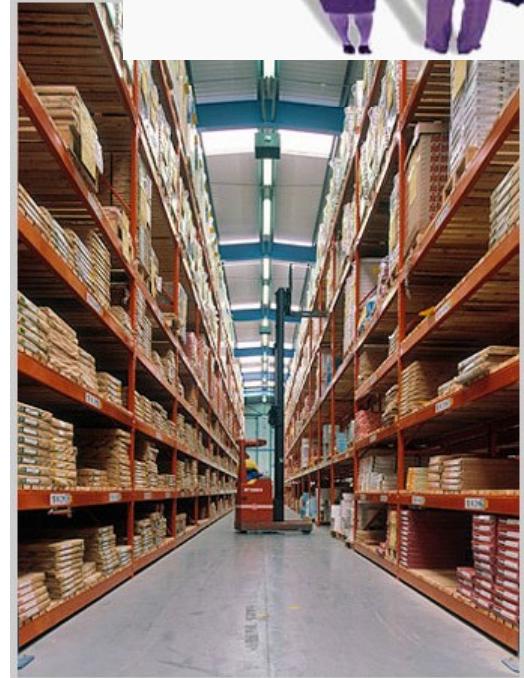
Data Warehouse Life Cycle & Architecture

Summary – last week

- Last week:
 - Assessment 01

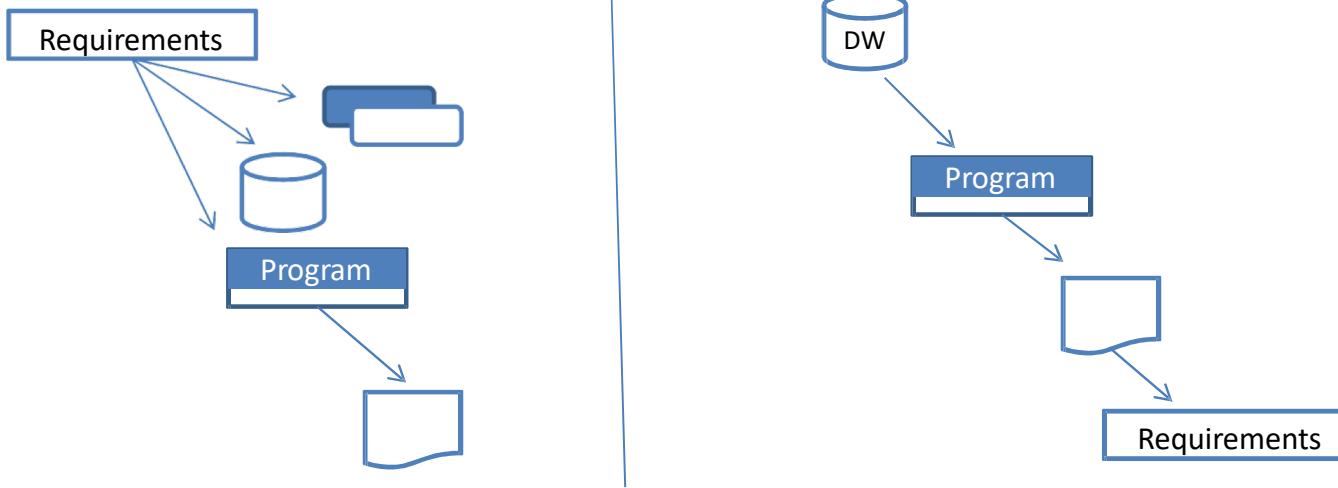


- This week:
 - DW Lifecycle and Architecture



Life cycle of DW

- Classical SDLC vs. DW SDLC

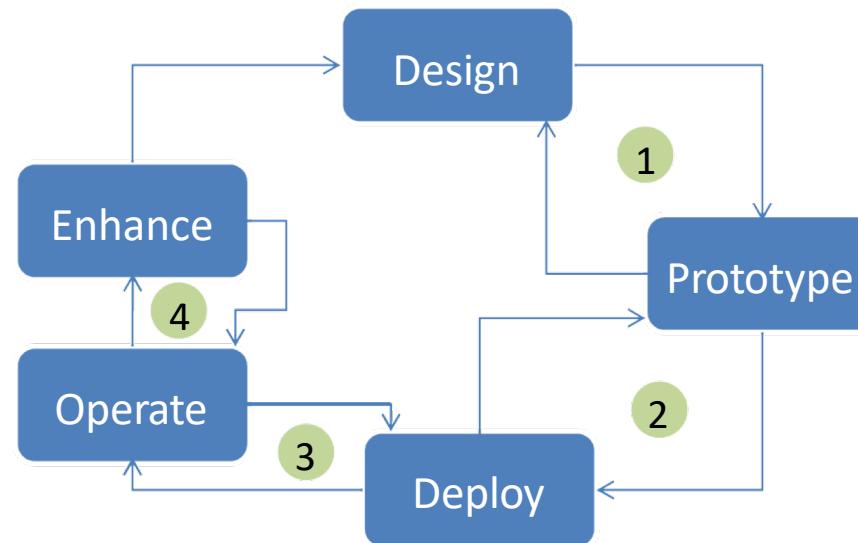


– DW SDLC is almost the opposite of classical SDLC

Life cycle of DW (cont'd.)

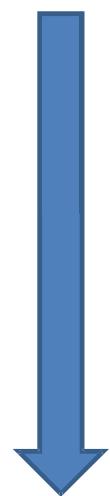
– Enhancement needs the modification of

- HW - physical components
- Operations and management processes
- Logical schema designs



Life cycle of DW (cont'd.)

- Classical SDLC vs. DW SDLC



Classical SDLC	DW SDLC
Requirements gathering	Implement warehouse
Analysis	Integrate data
Design	Test for bias
Programming	Program against data
Testing	Design DSS system
Integration	Analyze results
Implementation	Understand requirements

– Because it is the opposite of SDLC, DW SDLC is also called CLDS

Life cycle of DW (cont'd.)

- CLDS is a data driven development life cycle
 - It starts with data
 - Once data is at hand it is integrated and tested against bias
 - Programs are written against the data and the results are analyzed and finally the requirements of the system are understood
 - Once requirements are understood, adjustments are made to the design and the cycle starts all over
 - “spiral development methodology”



Operating a DW

- In Operating a DW the following phases can be identified
 - Monitoring
 - Extraction
 - Transforming
 - Loading
 - Analyzing



Monitoring

- Monitoring
 - Surveillance of the data sources
 - Identification of data modification which is relevant to the DW
 - Monitoring has an important role over the whole process deciding on which data the next steps will be applied on

Monitoring (cont'd.)

- Monitoring techniques
 - Active mechanisms – Event Condition Action (ECA)

EVENT	Payment
CONDITION	Account sum > 10 000 €
ACTION	Transfer to economy account

- Replication mechanisms
 - Snapshot:
 - Local copy of data, similar to a View
 - Used by Oracle 9i
 - Data replication
 - Replicates and maintains data in destination tables through data propagation processes
 - Used by IBM

Monitoring (cont'd.)

- Protocol based mechanisms
 - Since DBMS write protocol data for transaction management, the protocol can be used also for monitoring
 - Difficult due to the fact that the protocol format is proprietary and subject to change
- Application managed mechanisms
 - Hard to implement for legacy systems
 - Based on *time stamping* or *data comparison*

Extraction

- Extraction
 - Reads the data which was selected throughout the monitoring phase and inserts it in the data structures of the workplace
 - Due to large data volume, compression can be used
 - The time-point for performing extraction can be:
 - Periodical:
 - Weather or stock market information can be actualized more times in a day, while product specification can be actualized in a longer period of time

Extraction (cont'd.)

- On request:
 - For example when a new item is added to a product group
 - Event driven:
 - Event driven extraction can be helpful in scenarios where time, or the number of modifications over passing a specified threshold triggers the extraction. For example each night at 03:00 or each time 50 new modifications took place, an extraction is performed
 - Immediate:
 - In some special cases like the stock market it can be necessary that the changes propagate immediately to the warehouse
- The extraction largely depends on hardware and the software used for the DW and the data source

Transforming

- Transforming
 - Implies adapting data schema as well as data quality to the application requirements
 - Data integration:
 - Transformation in de-normalized data structures
 - Handling of key attributes
 - Adaptation of different types of the same data
 - Conversion of encoding:
 - “Buy”, “Sell” → I,2 vs. B,S → I,2
 - Correcting of wrong information

Transforming (cont'd.)

- Normalization:
 - “Michael Hill” → “Michael,Hill” vs.
“Hill Michael” → “Michael,Hill”
- Date handling:
 - “MM-DD-YYYY” → “MM.DD.YYYY”
- Measurement units and scaling:
 - 10 inch → 25,4 cm
 - 30 mph → 48,279 km/h
- Save calculated values
 - $\text{Price_incl_GST} = \text{Price_excl_GST} * 1.15$
- Aggregation
 - Daily sums can be added into weekly ones
 - Different levels of granularity can be used

Transforming (cont'd.)

- Data cleaning:
 - Consistency check
 - Delivery_date should be after Order_date
 - Completeness
 - Management of missing values as well as NULL values

Loading

- Loading
 - Loading usually takes place during weekends or nights when the system is not under user stress
 - Split between initial load to initialize the DW and the periodical load to keep the DW updated
 - Initial loading
 - Implies big volumes of data and for this reason a bulk loader is used
 - Usually performed by partitioning, parallelization and incremental actualization



Analyzing

- **Analyze**
 - Data access
 - Useful for extracting goal oriented information:
 - How many iPhones 7 were sold by Apple stores in Auckland in the last 3 calendar weeks of 2017?
 - Although it is a common OLTP query, it might be too complex for the operational environment to handle
 - OLAP
 - Used to analyze data contained in DW
 - Used to answer requests like:
 - In which district does a product group register the highest profit
 - How did the profit change in comparison to the previous month?

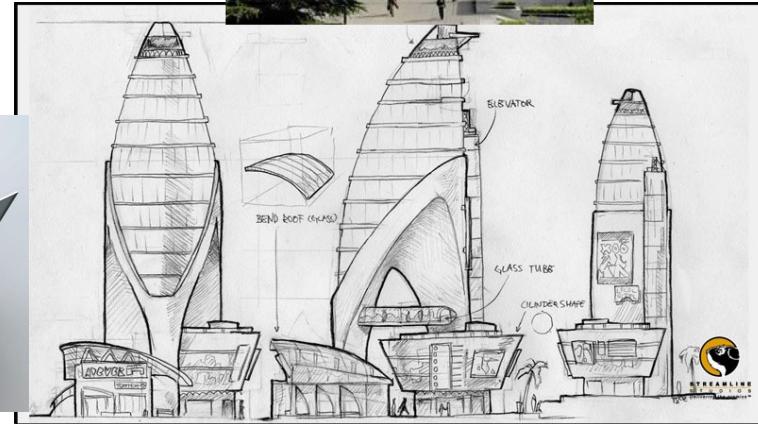
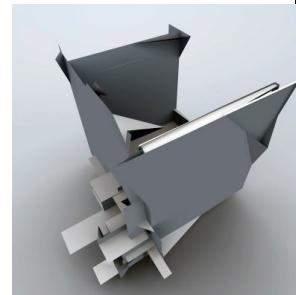
Analyzing (cont'd.)

- Mostly known as organized on a multidimensional data model
- Common operations for analyze are:
 - » Pivoting/Rotation
 - » Roll-up, Drill-down and Drill-across
 - » Slice and Dice
- Data mining
 - Useful for identifying hidden patterns
 - Refers to two separate processes:
 - KDD (Knowledge Discovery in Databases)
 - Prediction
 - Useful for answering questions like:
 - How did the sales of this product group evolve?
 - Methods and procedures for *data mining*
 - Clustering, Classification, Regression, Association rule learning

Architecture

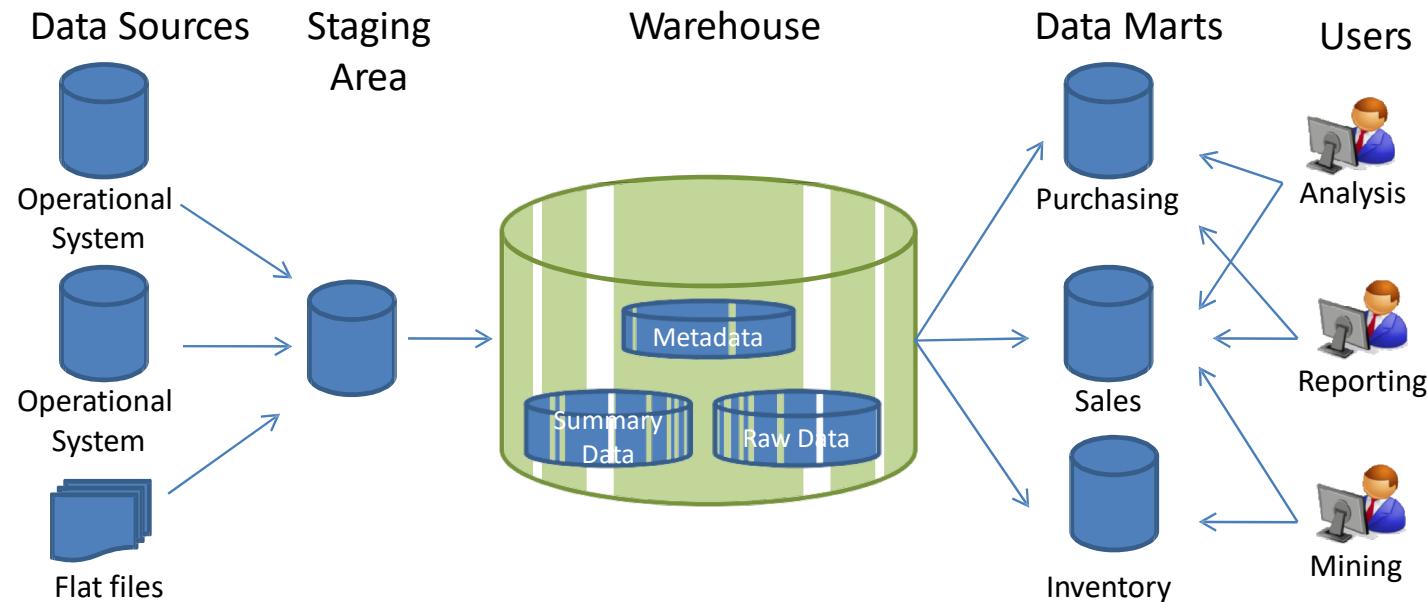
Architecture

1. Basic Architecture
2. Storage Structures
3. Tier Architectures
4. Distributed DW
5. DW Data Modeling



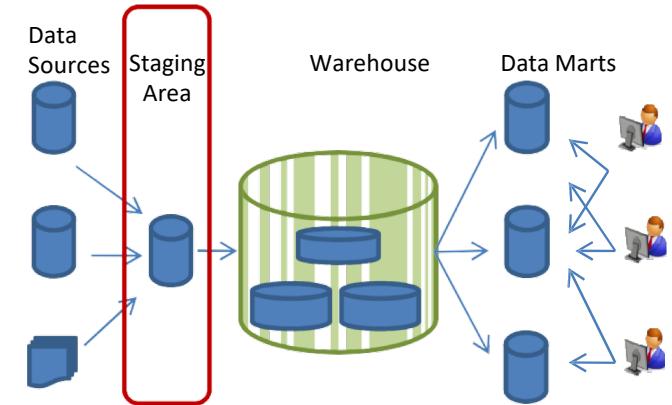
Basic Architecture

- Architecture of a DW



Basic Architecture (cont'd.)

- The Data Staging Area
 - Is both a storage and process area (the ETL process)
 - It represents everything that happens between the operational source system and the data presentation area
 - The key architectural requirement for data staging area is that it is off-limits to business users and does not provide query and presentation services



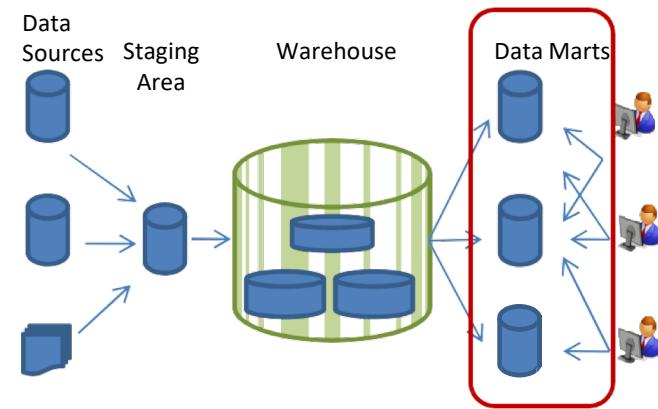
Basic Architecture (cont'd.)

- Customers aren't invited to visit the kitchen...
 - Similar to a restaurant's kitchen, the data staging area should be accessible only to skilled professionals



Basic Architecture (cont'd.)

- The Data Presentation Area
 - Is where data is organized, stored and made available for queries, report writers, and other analytical processing
 - This area is the Warehouse as far as the business community is concerned



Storage Structure

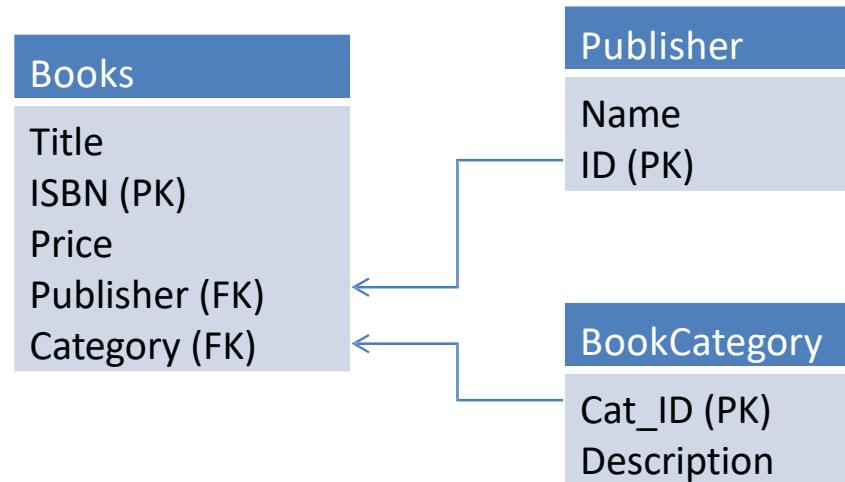
- Storage structure
 - After extraction from the operational data, in DW information is stored in databases
 - The databases are operated by a DBMS
 - Different database structures can be used for a DW:
 - Relational model (RDB) operated by a RDBMS
 - MultiDimensional model (MDB) operated by a MDBMS

Storage Structure (cont'd.)

- RDB and MDB are complementary and do not have to exclude each other
 - In the staging area some RDBMS can be used, however it must be off-limits to user queries because of performance reasons
 - By default, normalized databases are excluded from the presentation area, which should be strictly multi-dimensionally (MDBMS)

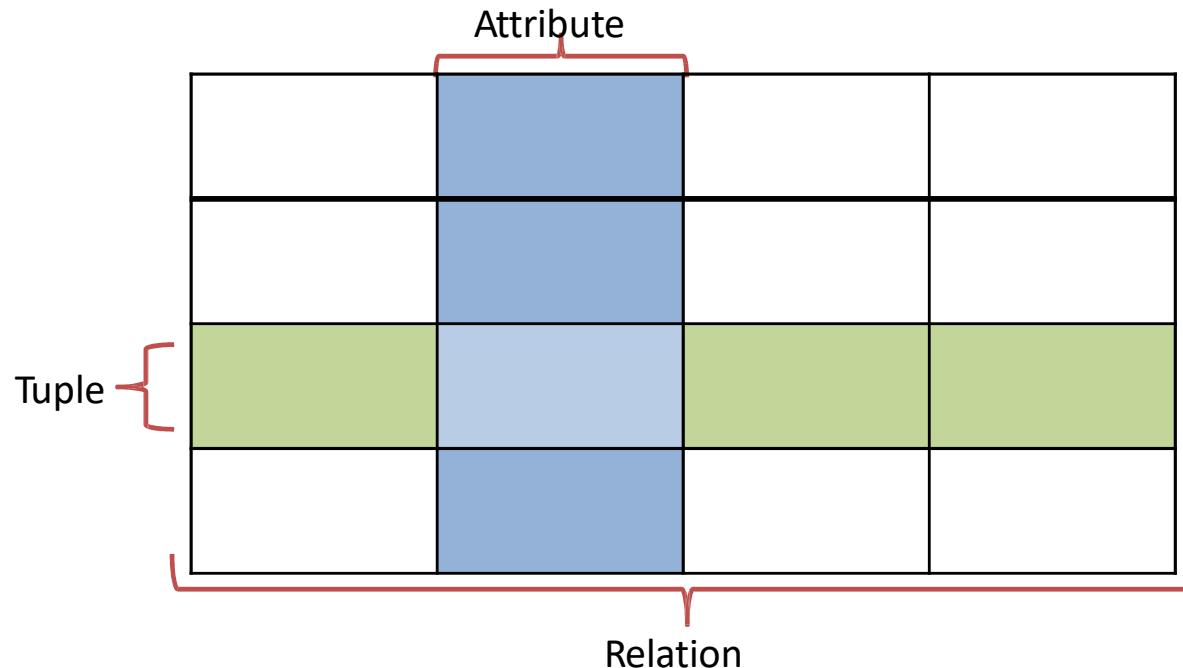
Relational DB

- DB in relational model
 - A database is seen as a collection of predicates over a finite set of variables
 - The content of the DB is modeled as a set of relations in which all predicates are satisfied



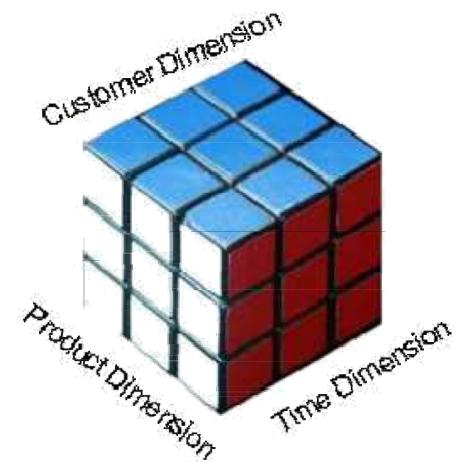
Relational DB (cont'd.)

- A relation is defined as a set of tuples that have the same attributes
 - It is usually described as a table



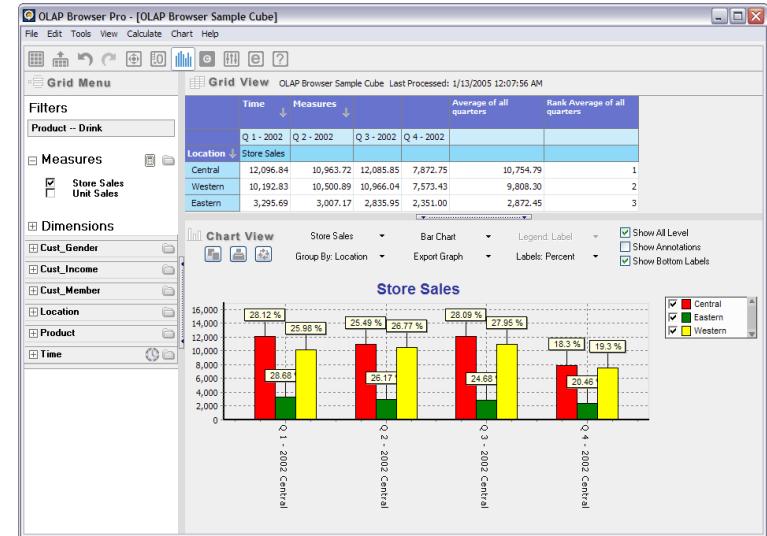
Multidimensional DB

- Multidimensional DB (MDB) are optimized for DW and OLAP applications
 - They are created using input from the staging area
 - Designed for efficient and convenient storage and retrieval of large volumes of data
 - Stored, viewed and analyzed from different perspectives called dimensions



Multidimensional DB (cont'd.)

- Example: an automobile manufacturer wants to increase sale volumes
 - Evaluation requires to view historical sale volume figures from multiple dimensions
 - Sales volume by model, by color, by dealer, over time



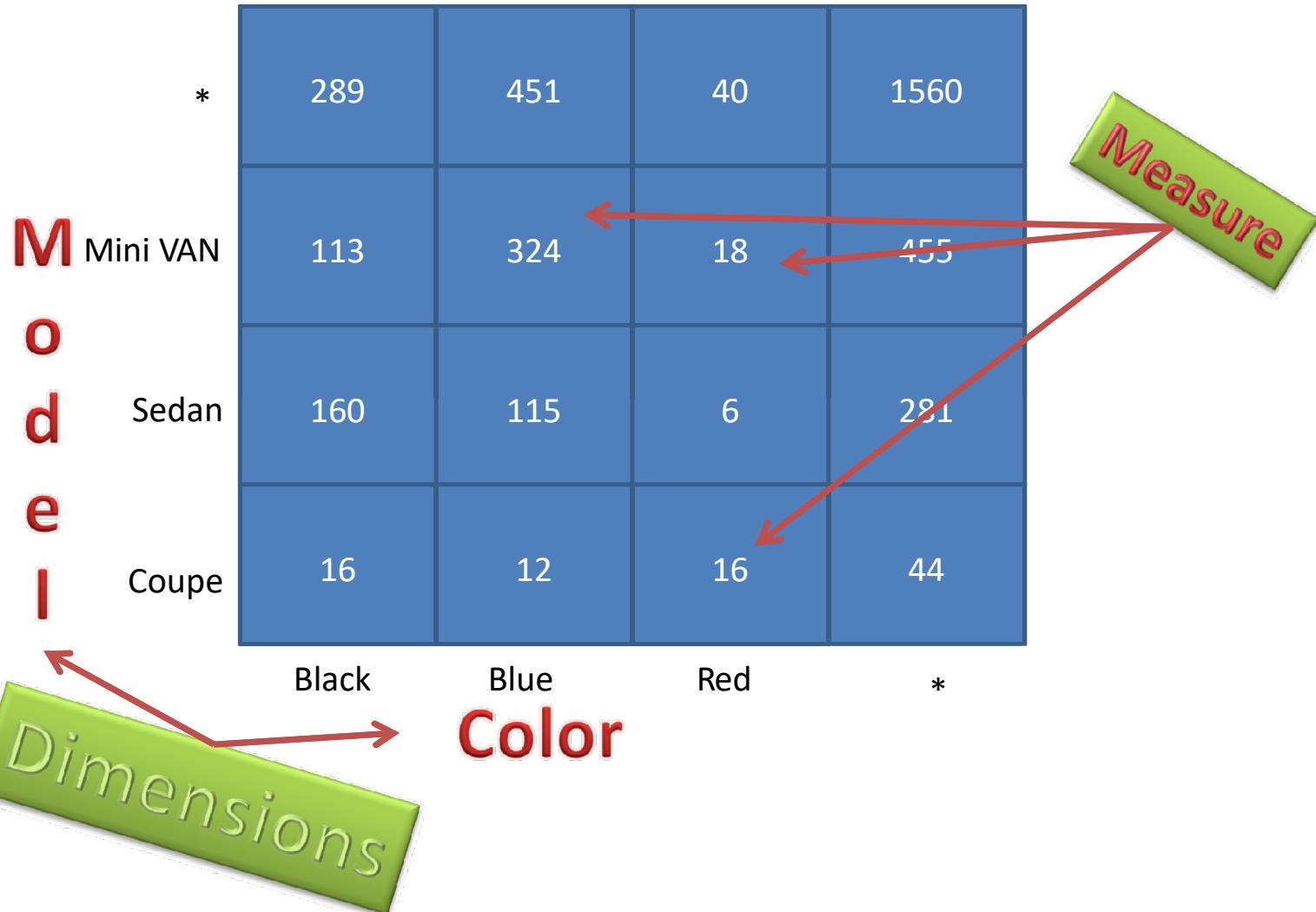
Multidimensional DB (cont'd.)

- A relational structure of the given evaluation would be

Model	Color	Sales volume
Mini VAN	Blue	324
Mini VAN	Black	113
Mini VAN	Red	18
Sedan	Black	160
Sedan	Blue	115
Sedan	Red	6
Sports coupe	Red	16
Sports coupe	Black	16
Sports coupe	Blue	12

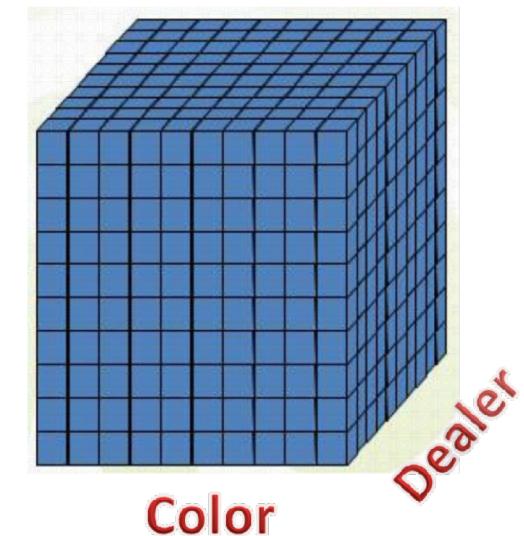
Multidimensional DB (cont'd.)

- Structure



Multidimensional DB (cont'd.)

- The complexity grows quickly with the number of dimensions and the number of positions
 - Example: 3 dimensions with 10 values each and no indexes
 - If we consider viewing information in a RDB it would result in a worst case of $10^3 = 1000$ records view

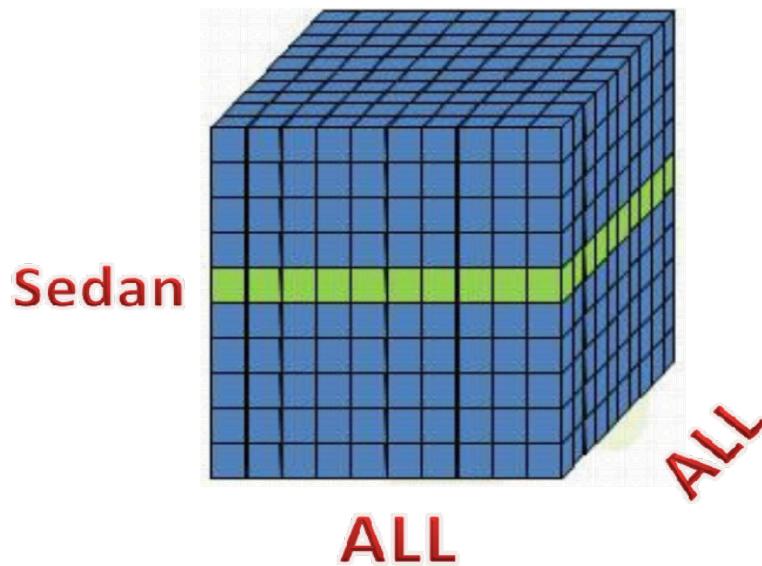


Multidimensional DB (cont'd.)

- Now, if we consider performance
 - For responding to a query when car type = Sedan, color = Blue, and dealer = Berg
 - RDBMS has to search through 1000 records to find the right record
 - MDB has more knowledge about where data lies
 - The maximum of searches in the case of MDB is of 30 positions

Multidimensional DB (cont'd.)

- If the query is more relaxed
 - Total sales across all dealers for all colors when car type = sedan
 - RDBMS still has to go through the 1000 records
 - MDB, however, goes only through a slice of 10x10



Multidimensional DB (cont'd.)

- Performance advantages
 - MDBs are an order of magnitude faster than RDBMSs
 - Performance benefits are more for queries that generate cross-tab views of data (the case of DW)
- Conclusion
 - The performance advantages offered by MDBs facilitates the development of interactive decision support applications like OLAP that can be impractical in a relational environment



RDB vs. MDB

- Any database manipulation is possible with both technologies
- MDBs however offer some advantages in the context of DW:
 - Ease of data presentation
 - Ease of maintenance
 - Performance



RDB vs. MDB (cont'd.)

- Ease of data presentation
 - Data views are natural output of the MDBs
 - Obtaining the same views in RDB requires a complex query
 - Example with Walmart and Sybase:
 - ```
select sum(sales.quantity_sold) from sales,products,product_categories,
manufacturers,stores,cities where manufacturer_name ='Colgate'
and product_category_name ='toothpaste'
and cities.population < 40 000
and trunc(sales.date_time_of_sale) = trunc(sysdate-1)
and sales.product_id = products.product_id
and sales.store_id = stores.store_id
and products.product_category_id = product_categories.product_category_id
and products.manufacturer_id = manufacturers.manufacturer_id
and stores.city_id = cities.city_id
```



# RDB vs. MDB (cont'd.)

- Ease of data presentation
  - Top k queries cannot be expressed well in SQL
    - Find the five cheapest hotels in Auckland
      - `SELECT * FROM hotels h WHERE h.city = Auckland AND 5 > (SELECT count(*) FROM hotels h1 WHERE h1.city = Auckland AND h1.price < h.price);`
    - Some RDBMS extended the functionality of SQL with STOP AFTER functionality
      - `SELECT * FROM hotels WHERE city = Auckland Order By price STOPAFTER 5;`

# RDB vs. MDB (cont'd.)

---

- Ease of maintenance
  - No additional overhead to translate user queries into requests for data
    - Data is stored as it is viewed
  - RDBs use indexes and sophisticated joins which require significant maintenance and storage to provide same intuitiveness

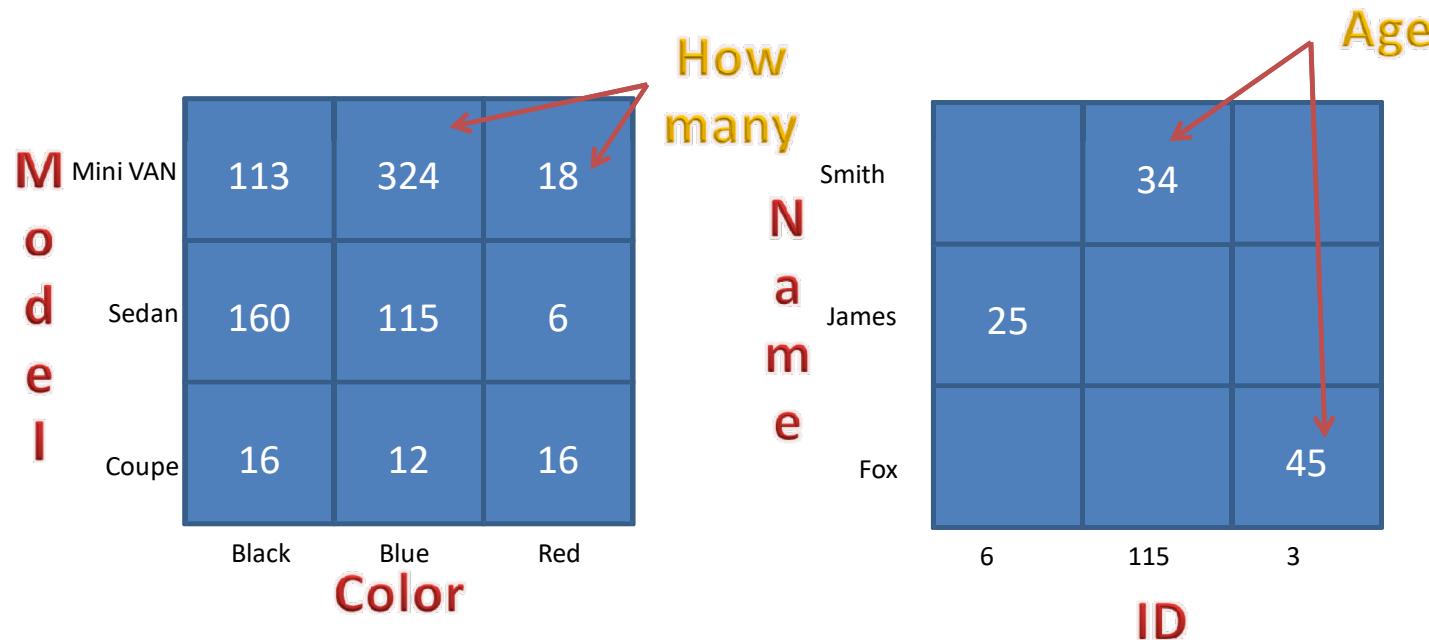
# RDB vs. MDB (cont'd.)

---

- Performance
  - – Performance of MDBs can be matched by RDBs through database tuning
  - – Not possible to tune the database for all possible ad-hoc queries
  - – Aggregate navigators are helping RDBs to catch up with MDBs as far as aggregation queries are concerned

# When MDBs are In-appropriate?

- When MDBs are in-appropriate?
  - If the dataset types are not highly related, using a MDB results in a sparse representation



# When MDBs are Appropriate?

- When MDBs are appropriate?
  - In the case of highly interrelated dataset types MDBs are recommended for greatest ease of access and analysis
  - Examples of applications
    - Financial Analysis and Reporting
    - Budgeting
    - Promotion Tracking
    - Quality Assurance and Quality Control
    - Product Profitability



# Summary

# Summary

- DW life cycle:
  - DW life cycle
  - Operating a DW
    - Monitoring
    - Extraction
    - Transforming
    - Loading
    - Analyzing
- DW architectures:
  - Basic architecture
  - Storage architecture

# Next lecture

---

- DW architecture (cont'd.)
  - Tier architecture
  - Distributed DW
  - DW data modelling