Data Warehouse

Multi-dimensional Data models

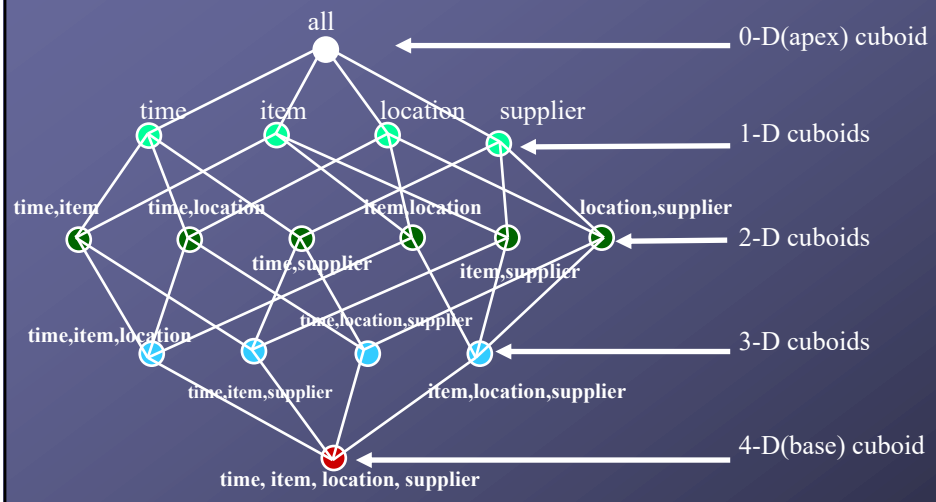
Learning Outcomes

- Define multi-dimensional data model
  - Star schema,
  - snowflake schema,
  - fat constellations
- Different Measures in Data Warehouse
- Different Data Warehouse Architectures

From Tables and Spreadsheets to Data Cubes

- A data warehouse is based on
  - multidimensional data model which views data in the form of a data cube
- A data cube allows data to be modeled and viewed in multiple dimensions (such as sales)
  - Dimension tables, such as item (item_name, brand, type), or time(day, week, month, quarter, year)
  - Fact table contains measures (such as Euros_sold) and keys to each of the related dimension tables
- Definitions
  - an n-Dimensional base cube is called a base cuboid
  - The top most 0-D cuboid, which holds the highest-level of summarisation, is called the apex cuboid
  - The lattice of cuboids forms a data cube

Cube: A Lattice of Cuboids

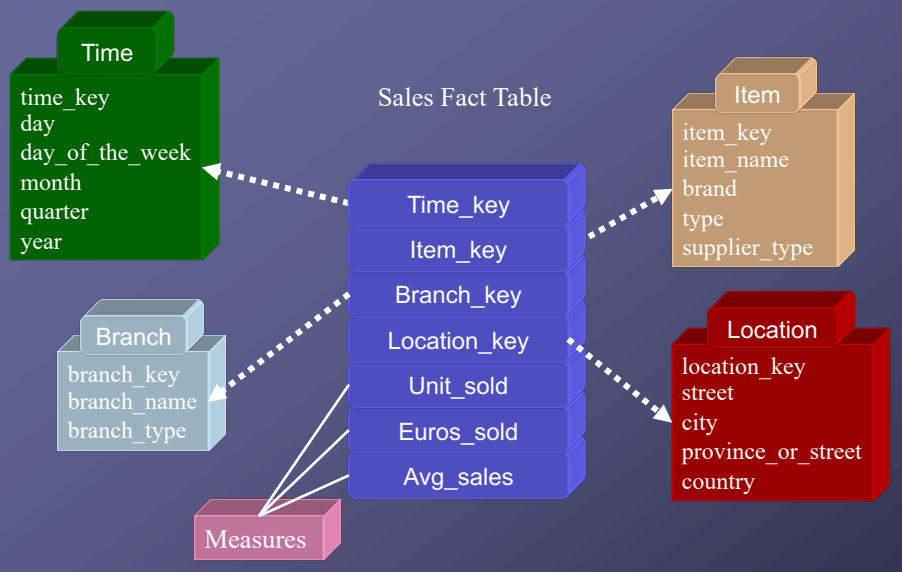


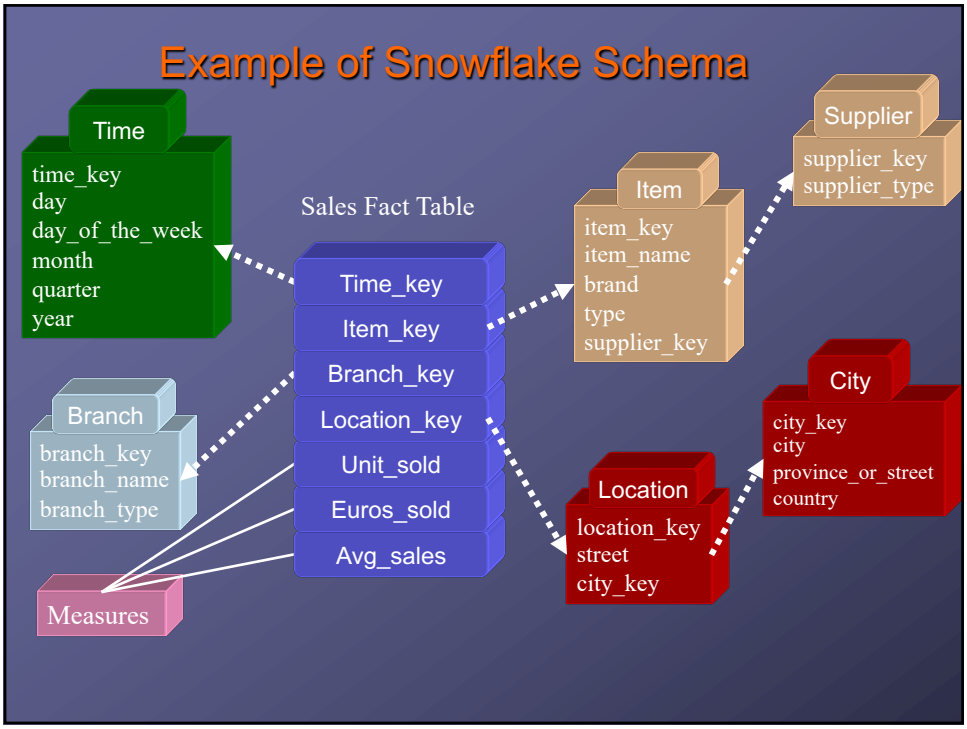
Conceptual Modeling of Data Warehouses

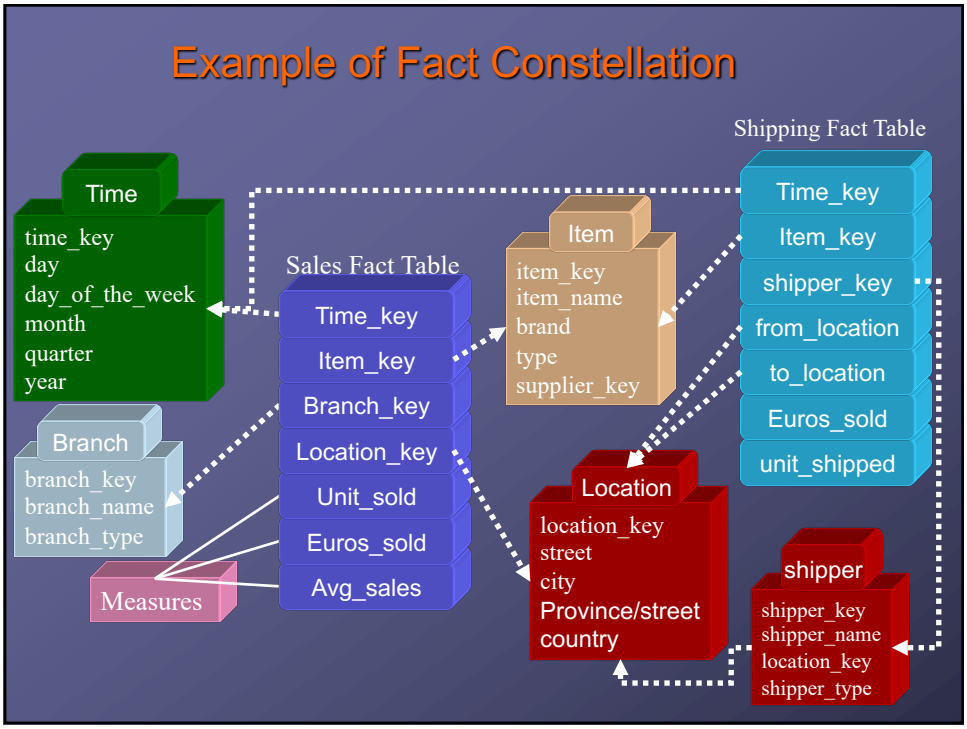
Modeling data warehouses: dimensions & measures

- Star schema
  - A fact table in the middle connected to a set of dimension tables
- Snowflake schema
  - A refinement of star schema where some dimensional hierarchy is normalised into a set of smaller dimension tables, forming a shape similar to snowflake
- Fact constellations
  - Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation

Example of Star Schema




Example of Snowflake Schema
Time
time_key
day
day_of_the_week
month
quarter
year
Sales Fact Table
Time_key
Item_key
Branch_key
Location_key
Unit_sold
Euros_sold
Avg_sales
Item
item_key
item_name
brand
type
supplier_key
Supplier
supplier_key
supplier_type
Branch
branch_key
branch_name
branch_type
Location
location_key
street
city_key
City
city_key
city
province_or_street
country
Measures


Example of Fact Constellation
Shipping Fact Table
Time
time_key
day
day_of_the_week
month
quarter
year
Sales Fact Table
Time_key
Item_key
Branch_key
Location_key
Unit_sold
Euros_sold
Avg_sales
Item
item_key
item_name
brand
type
supplier_key
Time_key
Item_key
shipper_key
from_location
to_location
Euros_sold
unit_shipped
Branch
branch_key
branch_name
branch_type
Location
location_key
street
city
Province/street
country
shipper
shipper_key
shipper_name
location_key
shipper_type
Measures

DMQL: Language Primitives

- Cube Definition (Fact Table)
  - define cube <cube_name> [<dimension_list>]: <measure_list>
- Dimension Definition (Dimension Table)
  - define dimension <dimension_name> as (<attribute_or_subdimension_list>)
- Special Case (Shared Dimension Tables)
  - First time as "cube definition"
  - define dimension <dimension_name> as <dimension_name_first_time> in cube <cube_name_first_time>

Defining a Star Schema in DMQL

```
define cube sales_star [time, item, branch, location]:
        Euros_sold = sum(sales_in_Euros),
        avg_sales = avg(sales_in_Euros),
        units_sold = count(*)
define dimension time as
        (time_key, day, day_of_week, month, quarter, year)
define dimension item as
        (item_key, item_name, brand, type, supplier_type)
define dimension branch as
        (branch_key, branch_name, branch_type)
define dimension location as
        (location_key, street, city, county, province, country)
```

Defining a Snowflake Schema in DMQL

```
define cube sales_snowflake [time, item, branch, location]:
        Euros_sold = sum(sales_in_Euros),
        avg_sales = avg(sales_in_Euros),
        units_sold = count(*)
define dimension time as
        (time_key, day, day_of_week, month, quarter, year)
define dimension item as
        (item_key, item_name, brand, type, supplier(supplier_key, supplier_type))
define dimension branch as
        (branch_key, branch_name, branch_type)
define dimension location as
        (location_key, street, city(city_key, county, province, country))
```

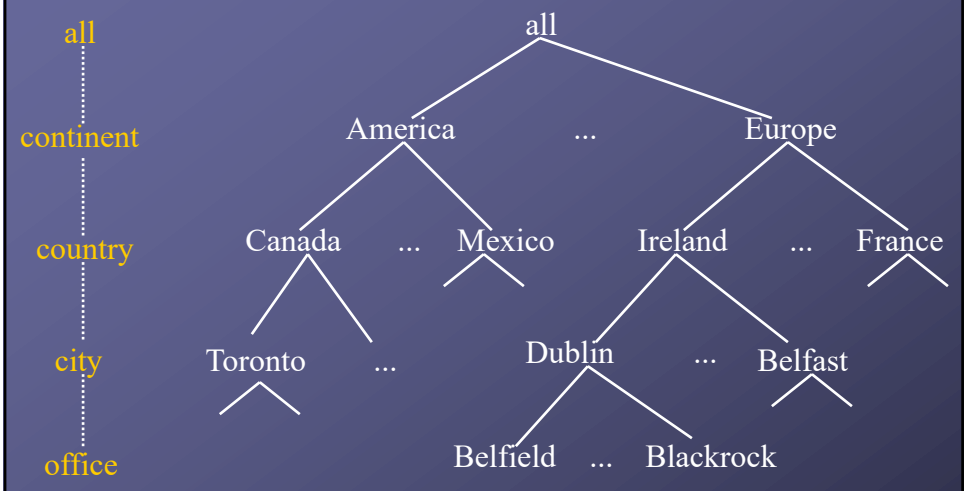
Defining a Fact Constellation in DMQL

```
define cube sales [time, item, branch, location]:
        Euros_sold = sum(sales_in_Euros), avg_sales = avg(sales_in_Euros),
        units_sold = count(*)
define dimension time as (time_key, day, day_of_week, month, quarter, year)
define dimension item as (item_key, item_name, brand, type, supplier_type)
define dimension branch as (branch_key, branch_name, branch_type)
define dimension location as (location_key, street, city, province_or_state,
        country)
define cube shipping [time, item, shipper, from_location, to_location]:
        Euro_cost = sum(cost_in_Euros), unit_shipped = count(*)
define dimension time as time in cube sales
define dimension item as item in cube sales
define dimension shipper as (shipper_key, shipper_name, location as location
    in cube sales, shipper_type)
define dimension from_location as location in cube sales
define dimension to_location as location in cube sales
```

Measures: Three Categories

- Distributive
  - if the result derived by applying the function to *n* aggregate values is the same as that derived by applying the function on all the data without partitioning.
    - E.g., count(), sum(), min(), max()
- Algebraic
  - if it can be computed by an algebraic function with *M* arguments (where *M* is a bounded integer), each of which is obtained by applying a distributive aggregate function.
    - E.g., avg(), min_N(), standard_deviation()
- Holistic
  - if there is no constant bound on the storage size needed to describe a sub-aggregate.
    - E.g., median(), mode(), rank()

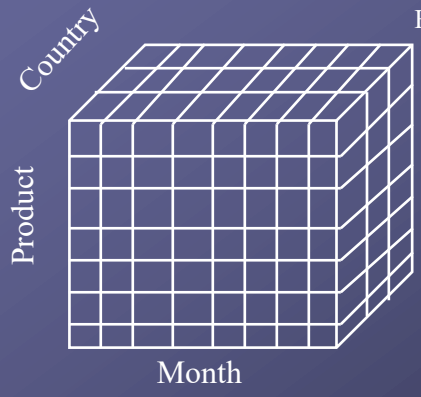
A Concept Hierarchy: Dimension (location)



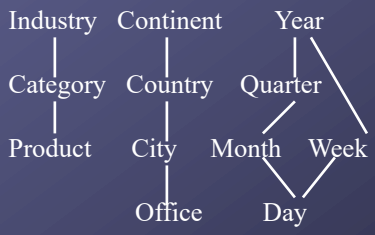
Concept hierarchy allows data to be handled at varying levels of abstractions.

Multidimensional Data

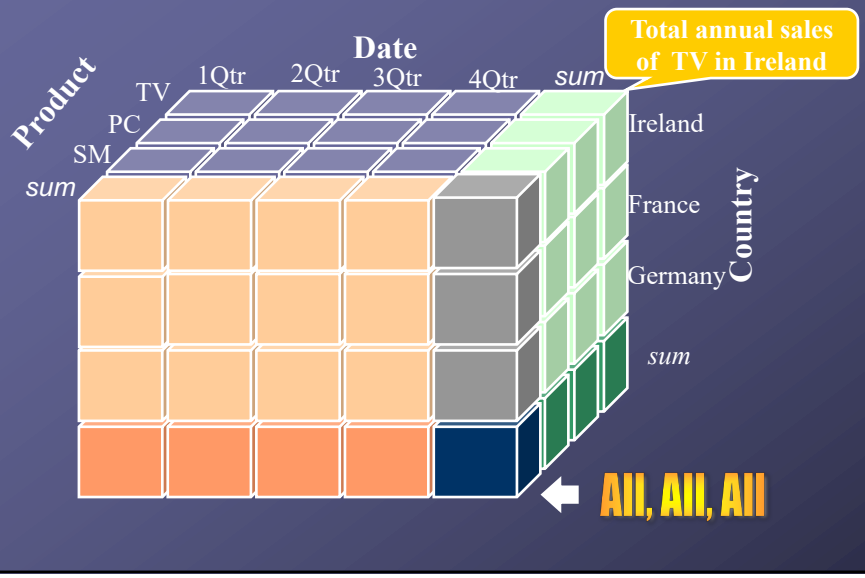
- Sales volume as a function of product, month, and Country



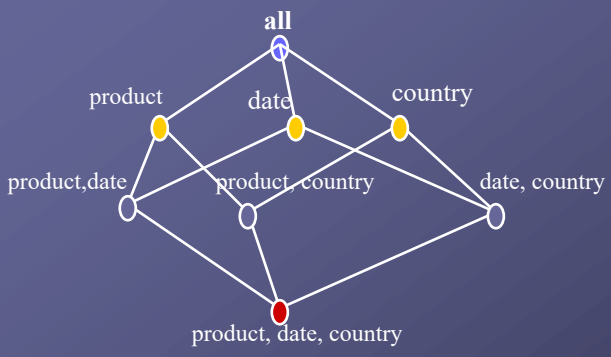
Dimensions: Product, Location, Time
Hierarchical summarisation paths



A Sample Data Cube



Cuboids Corresponding to the Cube



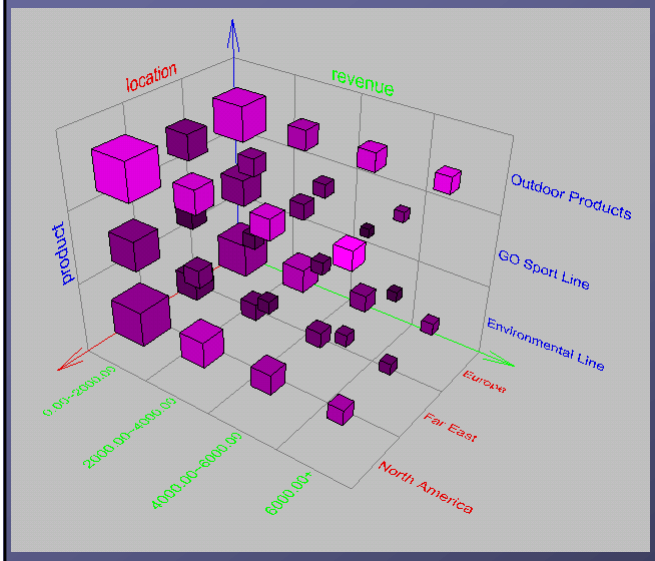
0-D(apex) cuboid

1-D cuboids

2-D cuboids

3-D(base) cuboid

Browsing a Data Cube

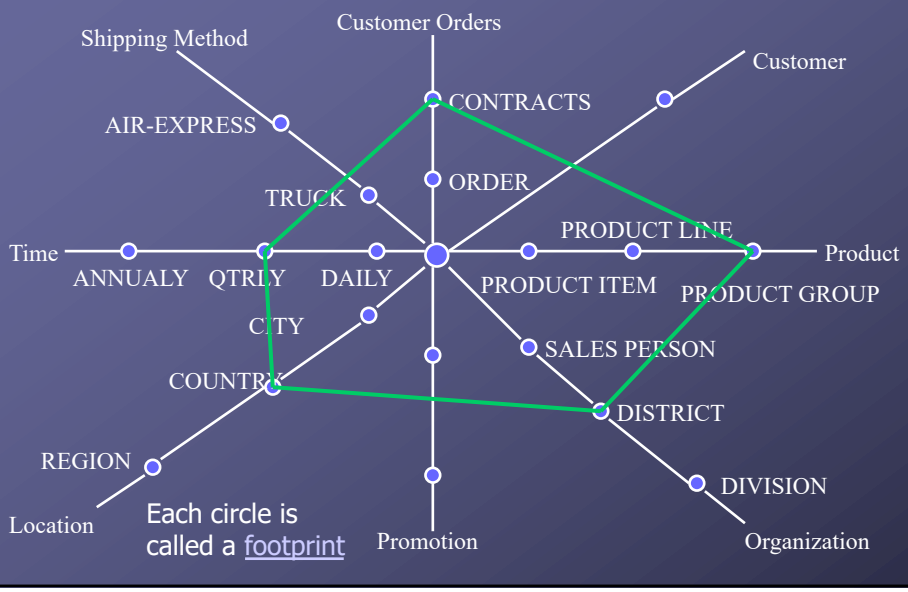


- Visualisation
- OLAP capabilities
- Interactive manipulation

Typical OLAP Operations

- Roll up (drill-up): summarise data
  - *by climbing up hierarchy or by dimension reduction*
- Drill down (roll down): reverse of roll-up
  - *from higher level summary to lower level summary or detailed data, or introducing new dimensions*
- Slice and dice
  - *project and select*
- Pivot (rotate)
  - *reorient the cube, visualisation, 3D to series of 2D planes.*
- Other operations
  - *drill across: involving (across) more than one fact table*
  - *drill through: through the bottom level of the cube to its back-end relational tables (using SQL)*

A Star-Net Query Model

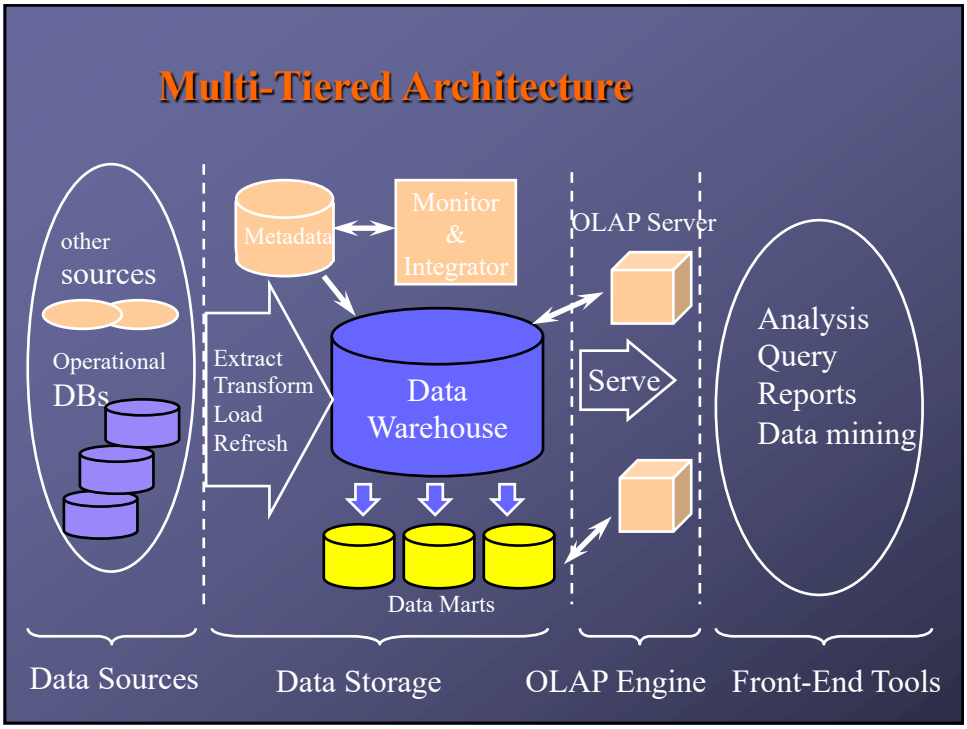


Design of a Data Warehouse: A Business Analysis Framework

- Four views regarding the design of a data warehouse
  - Top-down view
    - allows selection of the relevant information necessary for the data warehouse
  - Data source view
    - exposes the information being captured, stored, and managed by operational systems
  - Data warehouse view
    - consists of fact tables and dimension tables
  - Business query view
    - sees the perspectives of data in the warehouse from the view of end-user

Data Warehouse Design Process

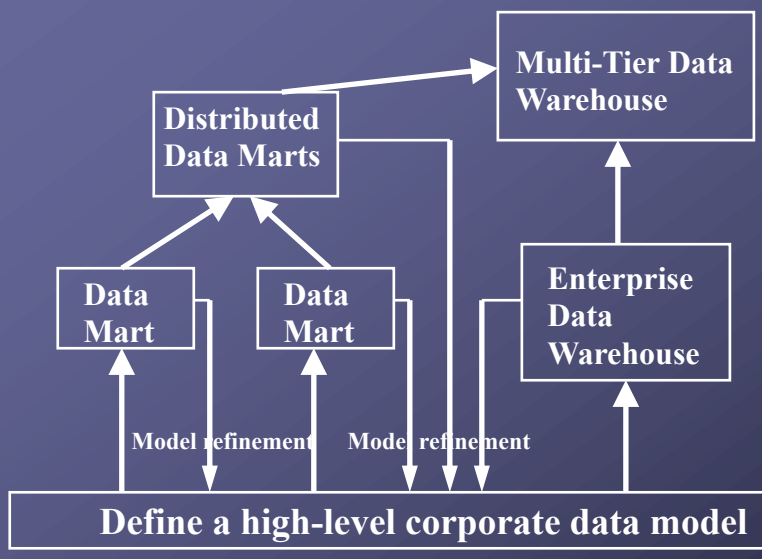
- Top-down, bottom-up approaches or a combination of both
  - Top-down: Starts with overall design and planning (mature)
  - Bottom-up: Starts with experiments and prototypes (rapid)
- From software engineering point of view
  - Steps: planning, data collection, DW design, testing and evaluation, DW deployment
  - Waterfall: structured and systematic analysis at each step before proceeding to the next
  - Spiral: rapid generation of increasingly functional systems, short turn around time, quick turn around
- Typical data warehouse design process
  - Choose a business process to model, e.g., orders, invoices, etc.
  - Choose the grain (*atomic level of data*) of the business process
  - Choose the dimensions that will apply to each fact table record
  - Choose the measure that will populate each fact table record



Three Data Warehouse Models

- Enterprise warehouse
  - collects all of the information about subjects spanning the entire organisation
- Data Mart
  - a subset of corporate-wide data that is of value to a specific group of users. Its scope is confined to specific, selected groups, such as marketing data mart
    - Independent vs. dependent (directly from warehouse) data mart
- Virtual warehouse
  - A set of views over operational databases
  - Only some of the possible summary views may be materialised

Data Warehouse Development: A Recommended Approach



OLAP Server Architectures

- Relational OLAP (ROLAP)
  - Use relational or extended-relational DBMS to store and manage warehouse data and OLAP middleware to support missing pieces
  - Include optimisation of DBMS backend, implementation of aggregation navigation logic, and additional tools and services
  - greater scalability
- Multidimensional OLAP (MOLAP)
  - Array-based multidimensional storage engine (sparse matrix techniques)
  - fast indexing to pre-computed summarised data
- Hybrid OLAP (HOLAP)
  - User flexibility, e.g., low level: relational, high-level: array
- Specialised SQL servers
  - specialised support for SQL queries over star/snowflake schemas

Exercise 1

Suppose that a data warehouse for a *Big University* consists of the following 4 dimensions: *students*, *module*, *semester*, and *lecturer* and 2 measures count and avg_grade. When at the lowest conceptual level (e.g., for a given student, module, semester, and lecturer combination), the avg_grade measure stores the actual module grade of the student. At higher conceptual levels, avg_grade stores the average grade for the given combination.

1. Draw a *snowflake schema* diagram for the data warehouse.
2. Starting with the base cuboid [*student*, *module*, *semester*, *lecturer*], what specific OLAP operations (e.g., roll-up from *semester* to year) should one perform in order to list the average grade of CS modules for each *Big University* student.
3. If each dimension has 5 levels (including all), such as "*student* < *major* < *status* < *university* < **all**", how many cuboids will this cube contain (including the base and apex cuboids)?