Data Mining and Data Warehousing 6

Data Warehouse Schema & OLAP Operations

Chittaranjan Pradhan
School of Computer Engineering,
KIIT University

Data Cubes

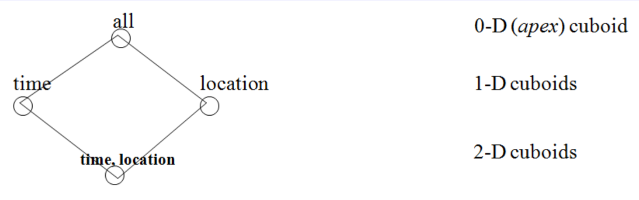
Data Cubes

- A data warehouse is based on a multidimensional data model which views data in the form of a data cube
- Data cube helps to arrange a complex data in a simple format
- Data cube represents the data along some measures of an interest
- It can be of 2-dimensional, 3-dimensional and higher dimensional
- Mainly used for the retrieval of the data
- It consists of categories of data called dimensions and measures
- Measure and dimension represents fact such as cost, time, locations

Conceptual Data Modeling/Lattice of Cuboids

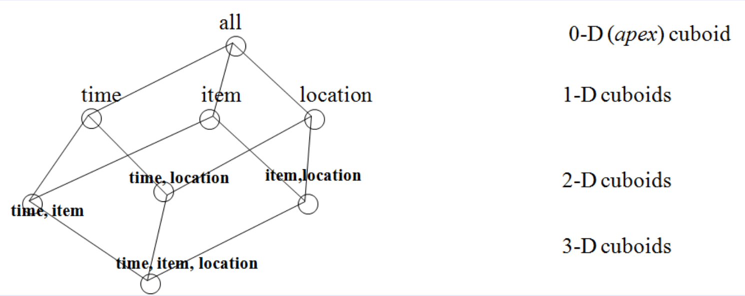
Conceptual Data Modeling/Lattice of Cuboids

- A *Cuboid* is the data cube at different degrees of summarization. Cuboids can be formed for each possible subset of dimensions which can form a lattice of cuboids
- In data warehousing, an n-D base cube is called a *base cuboid*. The top most 0-D cuboid, which holds the highest-level of summarization, is called the *apex cuboid*. The lattice of cuboids forms a *data cube*
- From a given set of dimensions, we can construct a lattice of cuboids, each showing the data at a different level of summarization



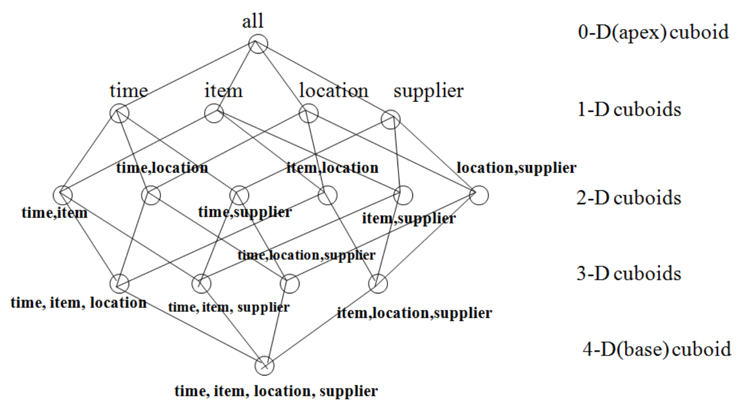
Conceptual Data Modeling/Lattice of Cuboids...

Conceptual Data Modeling/Lattice of Cuboids...



Conceptual Data Modeling/Lattice of Cuboids...

Conceptual Data Modeling/Lattice of Cuboids...



Data Warehouse Model Development

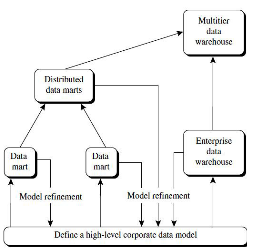
Data Warehouse Model Development

- Top-down Development
  - Advantages - Systematic solution and minimizes integration problem
  - Disadvantages - Difficult to achieve consistency and consensus for common data model, expensive, time taking task, not flexible
- Bottom-up Development
  - Design, development and deployment of independent data marts
  - Advantages - Fexible, low cost and rapid return of investment
  - Disadvantages - Difficult to integrate different data marts into a consistent data warehouse
- Solution: Develop in an incremental and evolutionary manner

Data Warehouse Model Development...

Data Warehouse Model Development...

- First: Define highlevel corporate data model that define the consistent and integrated view of different subjectes and (potential) usages
- Second: Independent data marts can be implemented in parallel with the enterprice warehouse
- Third: Distributed data marts can be constructed to integrate the data marts via hub server
- Finally: A multitier data warehouse is constructed where enterprice data warehouse is the sole custodian and is distributed to various dependent data marts



Data Warehouse Schema

Data Warehouse Schema

- A data warehouse requires a concise,subject-oriented schema (pictorial representation) that facilitates online data analysis
- 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 *normalized* 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

- *Every entry of a fact table can be represented as a dimension or as a measure such as cost, time and count (or quantity)*

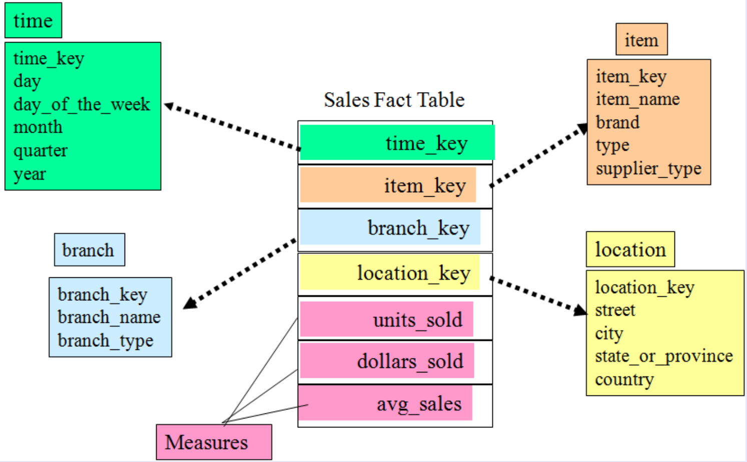
Star Schema

Star Schema

- Here, the center contains its fact table and the end points contain the dimension tables
- A star schema is a modeling paradigm in which the data warehouse contains a large, single, central fact table and a set of smaller dimension tables, one for each dimension
- The fact table contains the detailed summary data
- Its primary key has one key per dimension
- Each dimension is a single, highly Denormalized table
- There exists a 1:N relationship between the fact table and the dimension tables
- **Advantages**:
  - It is easy to understand, easy to define hierarchies, reduces the number of physical joins, requires low maintenance and very simple meta data
- **Disadvantages**:
  - Each dimension is represented by only one table, and each table contains a set of attributes. This constraint may introduce some redundancy

Star Schema...

Star Schema...



Snowflake Schema

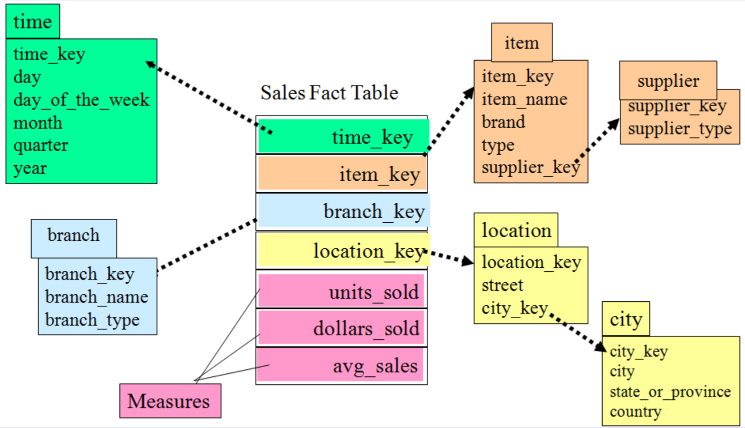
Snowflake Schema

- Here, there exists a single flake fact table and the dimension tables are either connected to the fact table or connected to themselves
- Snowflake schema consists of a single fact table and multiple dimension tables
- Dimension tables in a star schema are Denormalized, while those in a snowflake schema are normalized

- **Advantages**:
  - A normalized table is easier to maintain
  - Normalizing also saves storage space
- **Disadvantages**:
  - System performance may be adversely impacted due to larger number of join operations
  - Require more maintenance efforts because of the more lookup tables

Snowflake Schema...

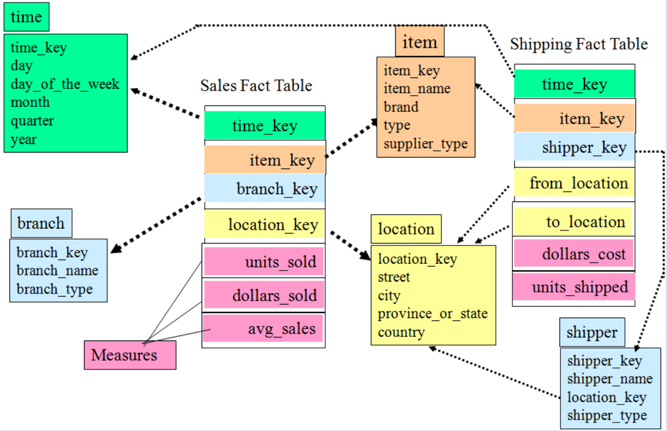
Snowflake Schema...



Fact Constellation

Fact Constellation

- Fact constellation is a kind of schema where we have more than one fact table sharing among them some dimension tables
- It is also called galaxy schema



Measures

Measures

- A data cube measure is a numerical function that can be evaluated at each point in the data cube space
- A measure value is computed for a given point by aggregating the data corresponding to the respective dimension - value pairs defining the given point
- **Distributive**:
  - An aggregate function is distributive if it can be computed in a distributed manner. Ex: Count(), Sum(), Min(), Max()
- **Algebraic**:
  - An aggregate function is algebraic if it can be computed by an algebraic function with ”m” arguments, each of which is obtained by applying a distributive aggregate function. Ex: Avg(), Min_N(), Max_N()
- **Holistic**:
  - An aggregate function is holistic if there is no constant bound on the storage size needed to describe a sub aggregate, i.e. there doesn’t exist an algebraic function with ”m” arguments that characterizes the computation. Ex: Median(), Mode(), Rank()

OLAP Operations

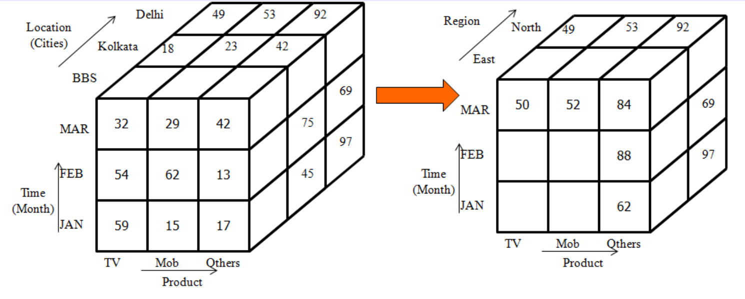
OLAP Operations

- OLAP is based on the multidimensional data model
- It allows managers and analysts to get an insight of the information through fast, consistent and interactive access to information
- It facilitates users to extract and present multidimensional data from different view
- It provides a user-friendly environment for interactive data analysis

Roll-Up/Drill-Up Operator

Roll-Up/Drill-Up Operator

- It causes an increase in data aggregation and removes a detail level from a hierarchy by climbing up hierarchy or by dimension reduction
- When roll up operation is performed one or more dimension is removed from the given cube
- $Roll - Up_{Location} C[Location, Month, Product] \rightarrow C[Region, Month, Product]$



Data Warehouse Schema & OLAP Operations

Chittaranjan Pradhan

Data Cubes

Conceptual Data Modeling/Lattice of Cuboids

Data Warehouse Model Development

Data Warehouse Schema

Star Schema

Snowflake Schema

Fact Constellation

Measures

OLAP Operations

Roll-Up/Drill-Up Operator

Drill-Down/Roll-Down Operator

Slice Operator

Dice Operator

Pivot Operator

OLAP Servers

DMQL

Star schema in DMQL

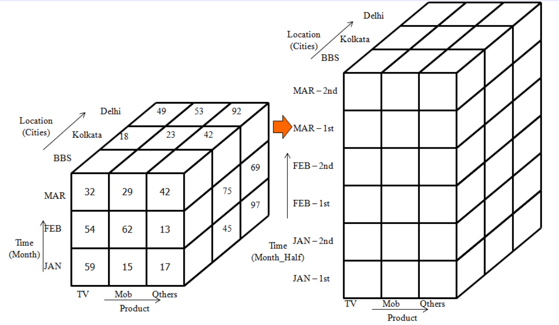
Snowflake schema in DMQL

Fact Constellation in DMQL

Drill-Down/Roll-Down Operator

Drill-Down/Roll-Down Operator

- It is complement to the roll - up operator from higher level summary to lower level summary or detailed data, or introducing new dimensions
- When drill-down is performed, one or more dimensions from the data cube are added
- $Drill - Down_{Month} C[Location, Month, Product] \rightarrow C[Location, Month\_Half, Product]$

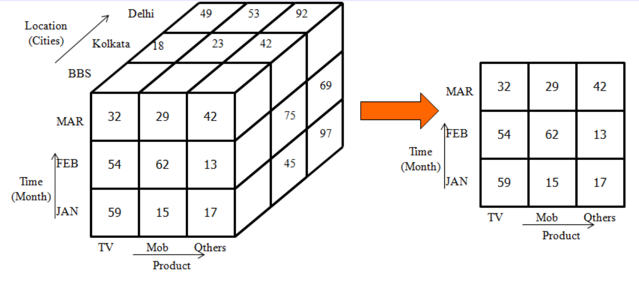


	TV	Mob	Others
MAR	32	29	42
FEB	54	62	13
JAN	59	15	17

Slice Operator

Slice Operator

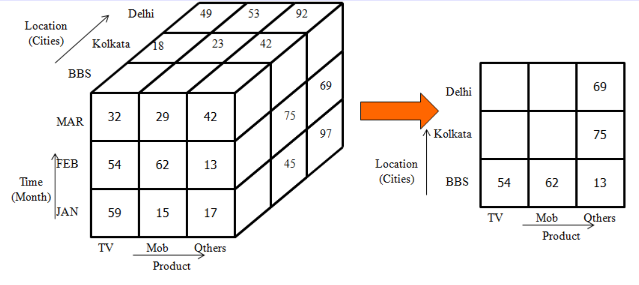
- Slicing reduces the number of cube dimensions after setting one of the dimensions to a specific value
- It reduce the dimensionality of the cubes
- $Slice_{Location='BBS'}\, C[Location, Month, Product] \rightarrow C[Month, Product]$



Slice Operator...

Slice Operator...

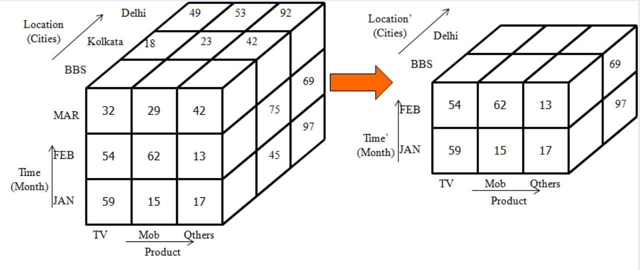
- $Slice_{Month='Feb'}\, C[Location, Month, Product] \rightarrow C[Location, Product]$



Dice Operator

Dice Operator

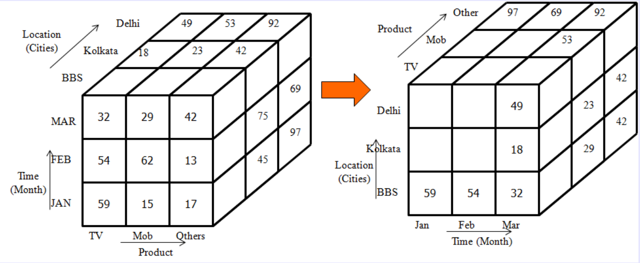
- Dicing reduces the set of data being analyzed by a selection criteria
- It selects two or more dimensions from a given cube and provides a new sub-cube
- $$Dice_{Month='Jan' OR 'Feb' AND Location='BBS' OR 'Delhi'} C[Location, Month, Product] \rightarrow C[Location', Month', Product]$$



Pivot Operator

Pivot Operator

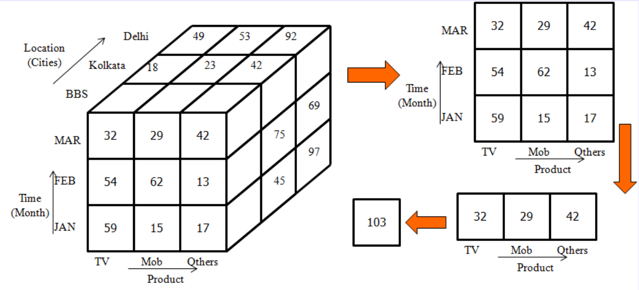
- It implies a change in layouts
- It rotates the data axis to view the data from different perspectives
- $Pivot_{90^0}\, C[Location, Month, Product] \rightarrow C[Product, Location, Month]$



OLAP Operations...

OLAP Operations...

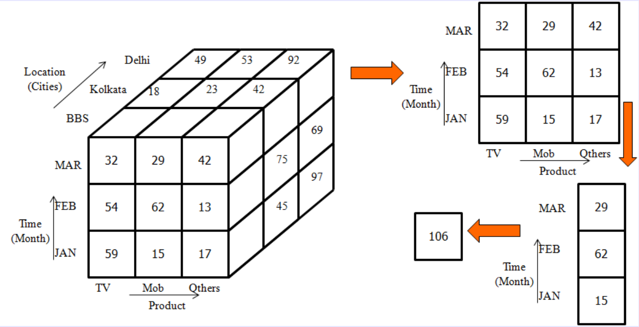
- Find out the total number of items sold in March in BBS
- $\sum(Slice_{Location='BBS'} + Slice_{Month='March'})$ or
- $\sum(Slice_{Month='March'}(Slice_{Location='BBS'}))$ or
- $Roll-Up_{Product}(Slice_{Location='BBS'} + Slice_{Month='March'})$ or
- $Roll-Up_{Product}(Slice_{Month='March'} + Slice_{Location='BBS'})$ or
- $Roll-Up_{Product}(Slice_{Month='March' AND Location='BBS'})$



OLAP Operations...

OLAP Operations...

- Find out the number of mobiles sold at BBS in all 3 months
- $\sum(Slice_{Location='BBS'} + Slice_{Product='Mob'})$



OLAP Servers

ROLAP (Relational OLAP)

- These servers are placed between relational back-end server and client front-end tools
- To store and manage the warehouse data, the relational OLAP uses relational or extended-relational DBMS

- Advantages
  - ROLAP servers can be easily used with existing RDBMS
  - ROLAP tools do not use pre-calculated data cubes
  - ROLAP server offers highly scalability
  - Can handle large amount of information

- Disadvantages
  - ROLAP needs high utilization of manpower, software and hardware resources
  - Query performance in this model is slow

OLAP Servers...

MOLAP (Multidimensional OLAP)

- MOLAP uses array-based multidimensional storage engines for multidimensional views of data
- With multidimensional data stores, the storage utilization may be low if the dataset is sparse

- Advantages
  - Fast information retrieval
  - Easier to use
  - Suitable for slicing and dicing operations
  - Capable of performing complex operations

- Disadvantages
  - MOLAP are not capable of containing detailed data
  - The storage utilization may be low if the data set is sparse

OLAP Servers...

HOLAP (Hybrid OLAP)

- HOLAP is a mixture of both ROLAP and MOLAP
- It offers fast computation of MOLAP and higher scalability of ROLAP
- HOLAP server allows to store large data volumes of detailed information

- Advantages
  - HOLAP provides the benefits of both ROLAP and MOLAP
  - It provides quick access at all levels of aggregation

- Disadvantages
  - HOLAP architecture is very complicated
  - There are higher chances of overlapping especially into their functionalities

DMQL

DMQL

- Cube definition : Fact table
  define cube <cube_name> [<dimension_list]: <measure_list>

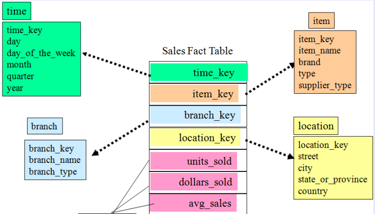
- Dimension definition : Dimension Table
  define dimension <dimension_name> as (<attribute_or_subdimension_list>)

- 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>

Star schema in DMQL

Star schema in DMQL

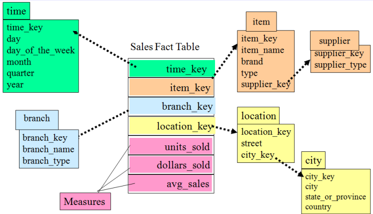
- *define cube sales_star [time, item, branch, location]: dollars_sold = sum(sales_in_dollars), avg_sales = avg(sales_in_dollars), units_sold = count(*)*
- *define dimension time as (time_key, day, day_of_week, month, quarter, year)*
- *define dimension item as (item_key, item_name, brand, type, supplier_type)*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



Snowflake schema in DMQL

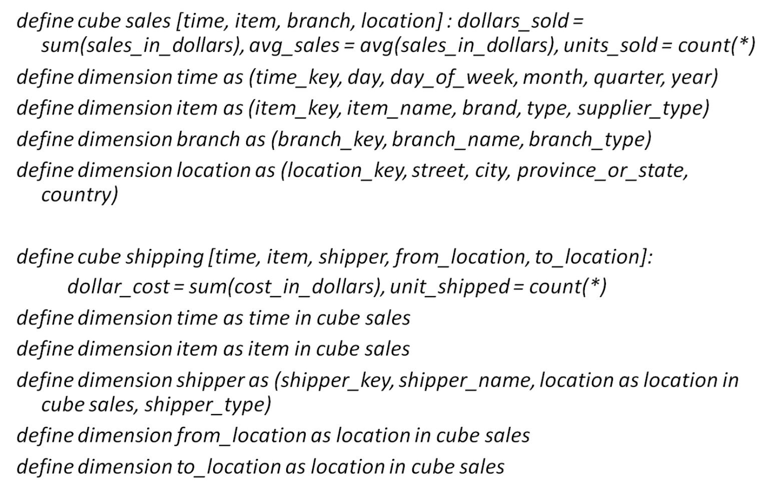
Snowflake schema in DMQL

- *define cube sales_snowflake [time, item, branch, location]: dollars_sold = sum(sales_in_dollars), avg_sales = avg(sales_in_dollars), 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, province_or_state, country))*



Fact Constellation in DMQL

Fact Constellation in DMQL

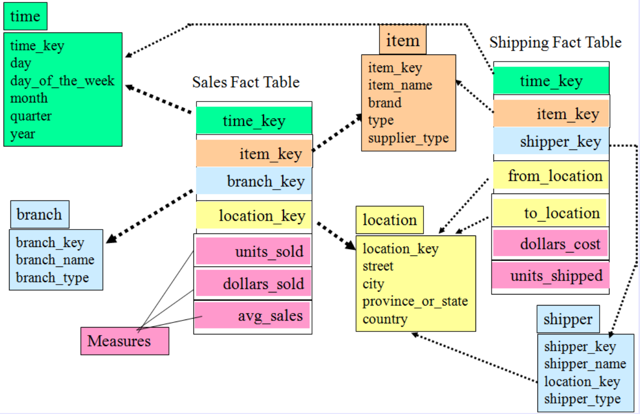


```
define cube sales [time, item, branch, location] : dollars_sold =
    sum(sales_in_dollars), avg_sales = avg(sales_in_dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week, month, quarter, year)
define dimension item as (item_key, item_name, brand, type, supplier_type)
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]:
        dollar_cost = sum(cost_in_dollars), 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
```

Fact Constellation in DMQL...

Fact Constellation in DMQL...


time
time_key
day
day_of_the_week
month
quarter
year
Sales Fact Table
time_key
item_key
branch_key
location_key
units_sold
dollars_sold
avg_sales
Measures
item
item_key
item_name
brand
type
supplier_type
Shipping Fact Table
time_key
item_key
shipper_key
from_location
to_location
dollars_cost
units_shipped
branch
branch_key
branch_name
branch_type
location
location_key
street
city
province_or_state
country
shipper
shipper_key
shipper_name
location_key
shipper_type