SGG 4653
Advance Database System

Data Warehouse

Contents

- Objectives of this topic:
  - To understand data warehouse
  - To know the architecture of data warehouse
  - To understand different schemas of data warehouse
- Contents of this topic:
  - Data Warehouse
  - Data Warehouse Architecture
  - Data Warehouse Life Cycle

What is Data Warehouse?

- "A data warehouse is a <u>subject-oriented</u>, <u>integrated</u>, <u>time-variant</u>, and <u>nonvolatile</u> collection of data in support of management's decision-making process."—W. H. Inmon

- Data warehousing:
  - The process of constructing and using data warehouses

What is Data Warehouse?

- Defined in many different ways, but not rigorously.
  - A decision support database that is maintained separately from the organization’s operational database
  - Support information processing by providing a solid platform of consolidated, historical data for analysis.

What is a data warehouse?

- A database filled with large volumes of cross-indexed historical business information that users can access with various query tools.
- The warehouse usually resides on its own server and is separate from the transaction-processing or “run-the-business” systems.

What is a data warehouse?

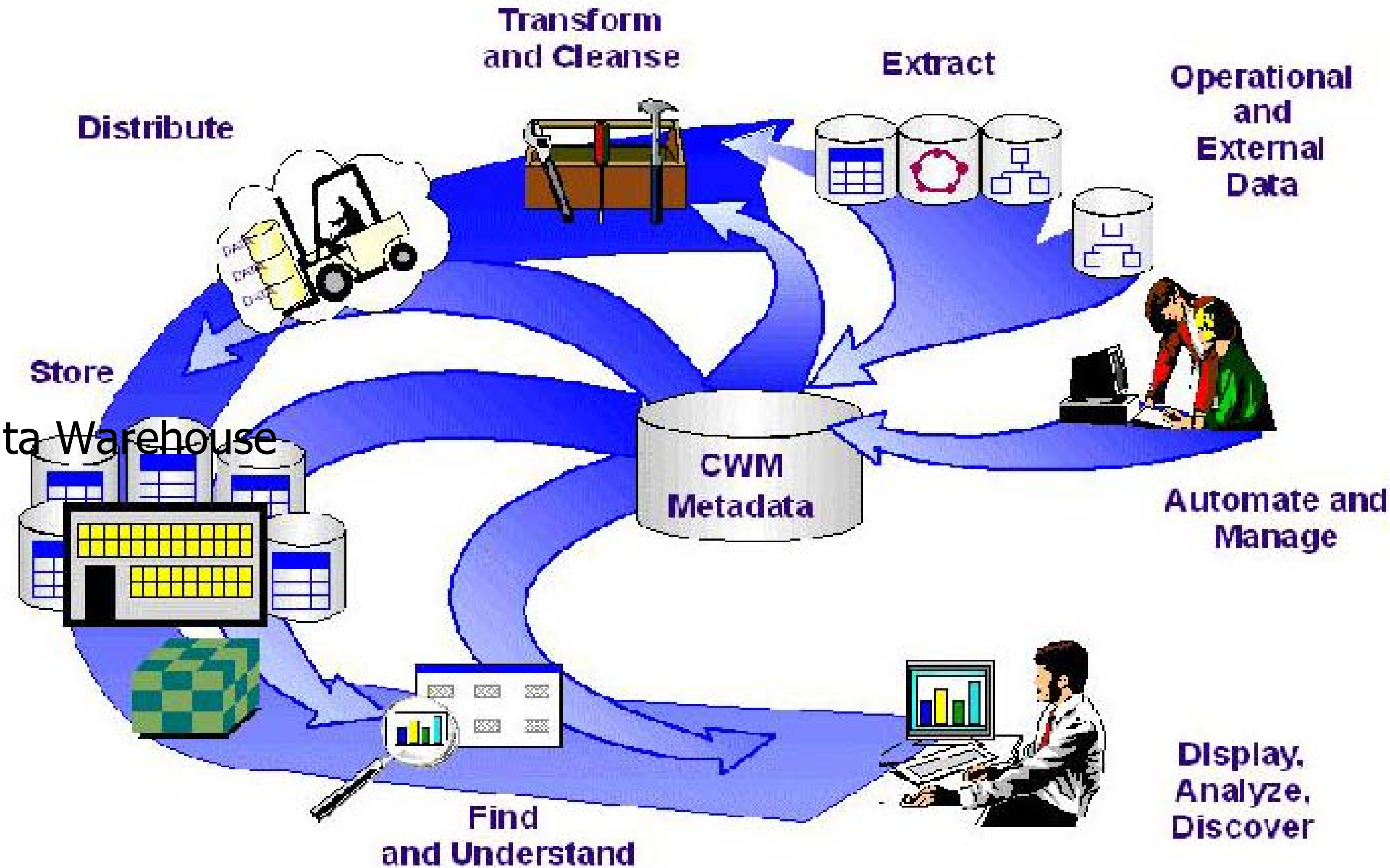
- It brings all the various sets of data together

  - Financial data
  - Personnel data
  - Building infrastructure data
  - Student demographic information
  - Student program information
  - Student achievement information

  - Example: Center for Educational Performance and Information's Michigan Education (CEPI) Information System.

(80% of work is data cleansing.)

Warehouse Scenario



CWM – Common Warehouse Meta Model

Data Warehouse—1) Subject-Oriented

- Organized around major subjects, such as customer, product, sales.
- Focusing on the modeling and analysis of data for decision makers, not on daily operations or transaction processing.
- Provide a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process.

Data Warehouse— 2) Integrated

- Constructed by integrating multiple, heterogeneous data sources
  - relational databases, flat files, on-line transaction records
- Data cleaning and data integration techniques are applied.
  - Ensure consistency in naming conventions, encoding structures, attribute measures, etc. among different data sources
    - E.g., Hotel price: currency, tax, breakfast covered, etc.
  - When data is moved to the warehouse, it is converted.

Data Warehouse—Integrated (cont'..)

- The Storage Managed by
  - <u>Relational databases</u>
    - like those from Oracle Corp. or Informix Software Inc.
  - Specialized hardware
    - symmetric multiprocessor (SMP)
    - or massively parallel processor (MPP) machines
- The majority of warehouse storage today is being managed by <u>relational databases running on Unix platforms</u>.
- Oracle, Sybase Inc., IBM Corp. and Informix control 65 percent of the warehouse storage market. *Meta Group Inc. (1996)*

Data Warehouse— 3) Time Variant

- The time horizon for the data warehouse is significantly longer than that of operational systems.
  - Operational database: current value data.
  - Data warehouse data: provide information from a historical perspective (e.g., past 5-10 years)
- Every key structure in the data warehouse
  - Contains an element of time, explicitly or implicitly
  - But the key of operational data may or may not contain "time element".

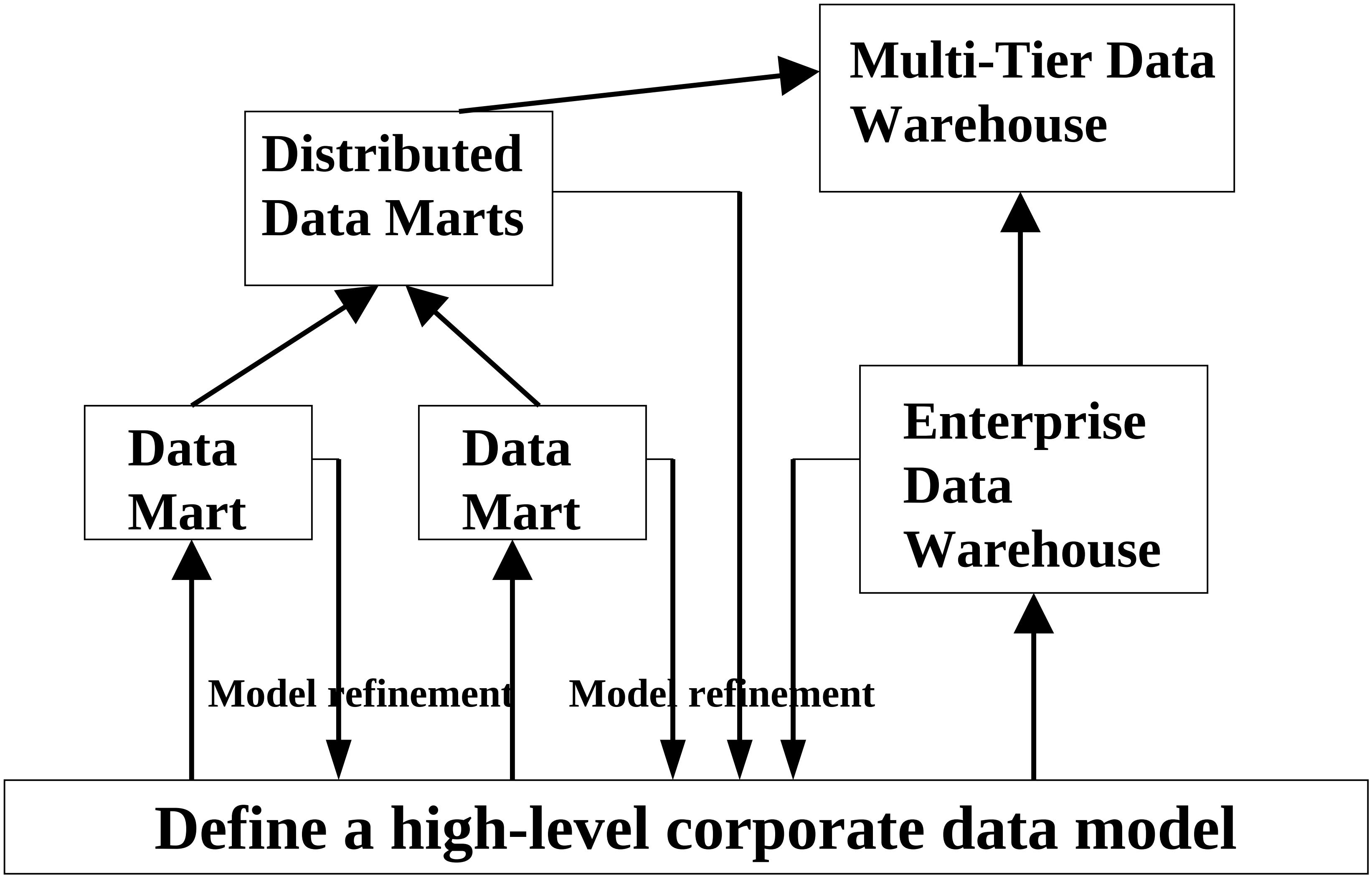
Data Warehouse— 4) Non-Volatile

- A physically separate store of data transformed from the operational environment.
- Operational update of data does not occur in the data warehouse environment.
  - Does not require transaction processing, recovery, and concurrency control mechanisms
  - Requires only two operations in data accessing:
    - *initial loading of data* and *access of data*.

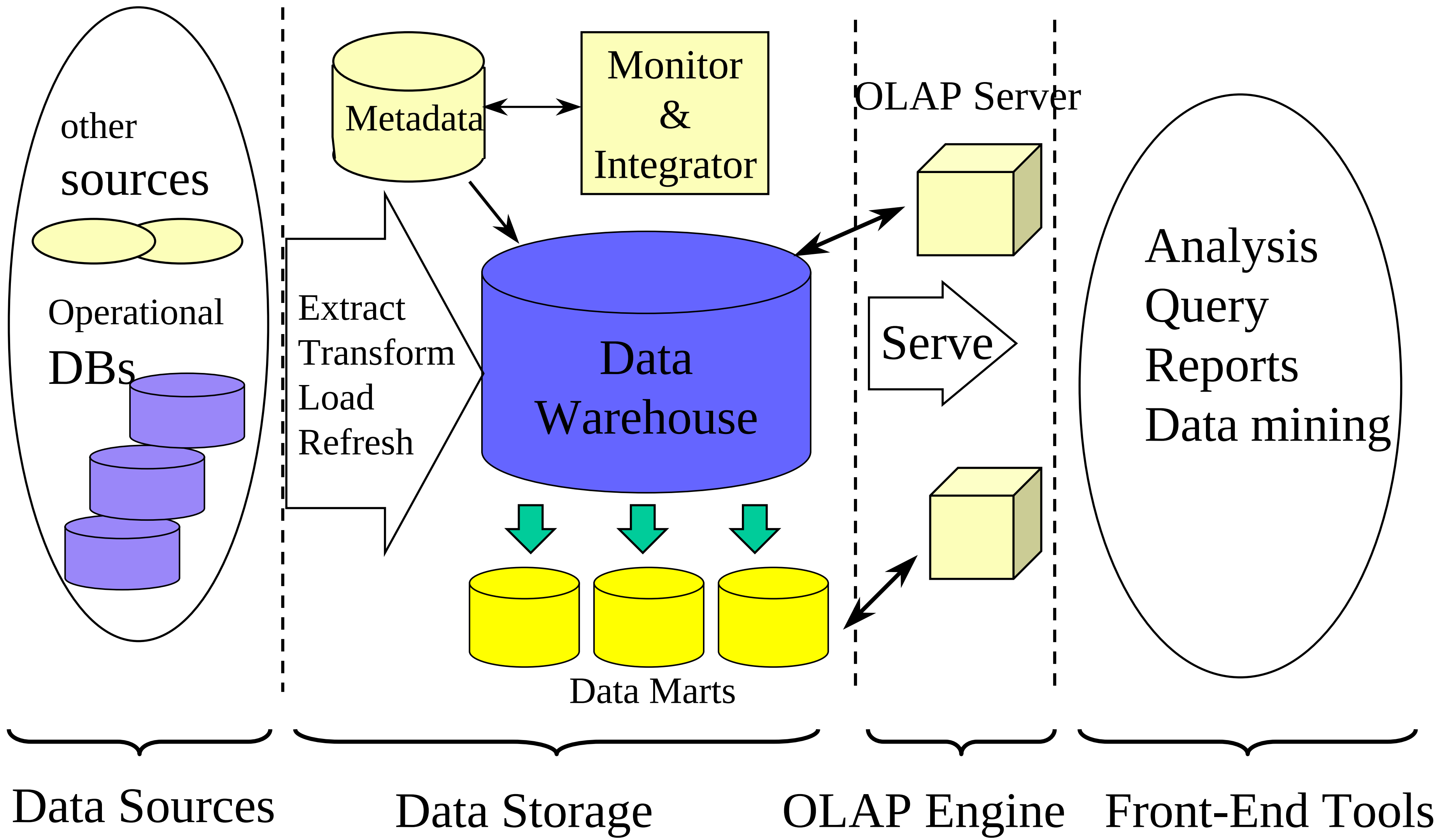
Three Data Warehouse Models

- Enterprise warehouse
  - collects all of the information about subjects spanning the entire organization
- Data Mart
  - a subset of corporate-wide data that is of value to a specific groups 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 materialized

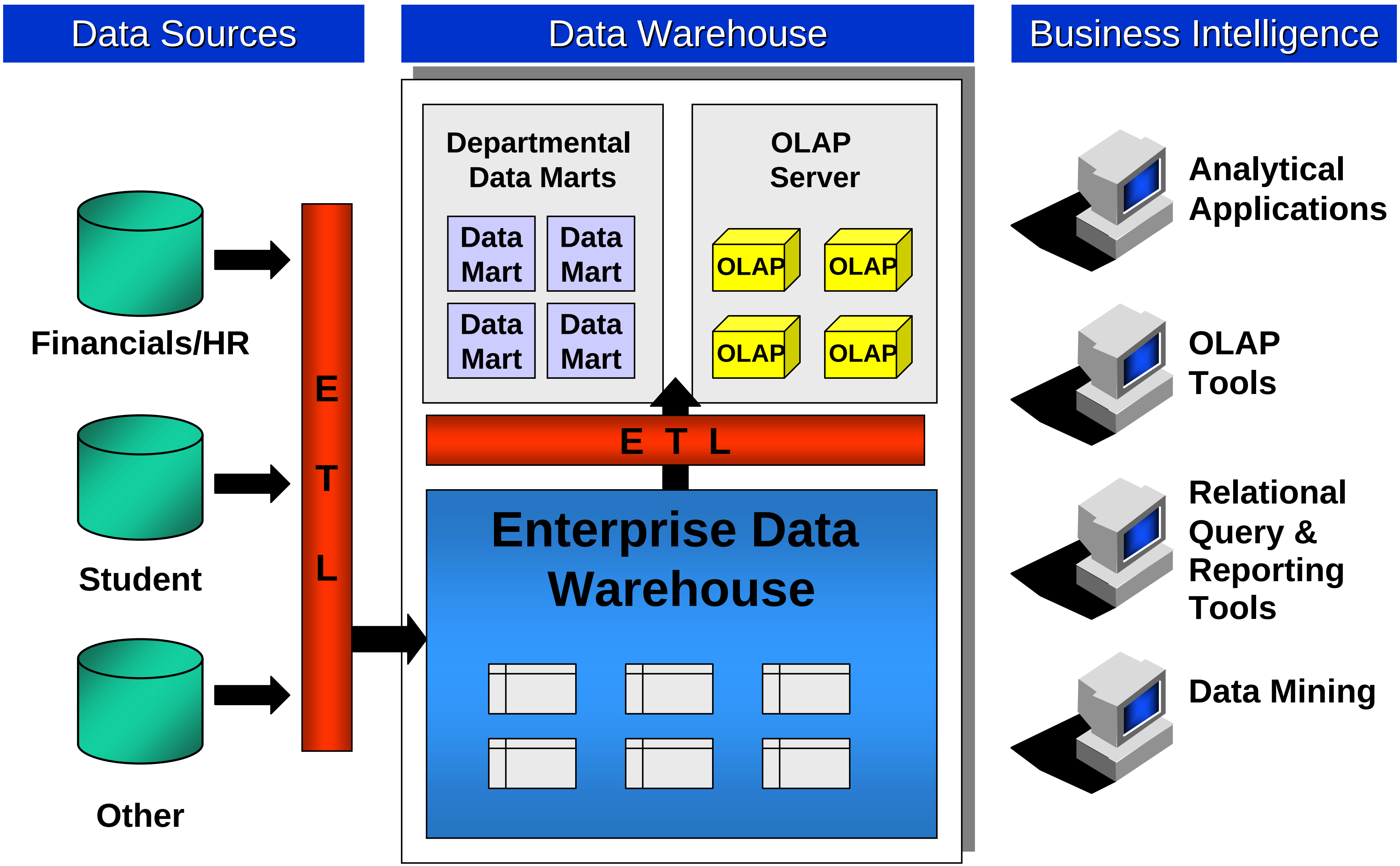
Data Warehouse Development: A Recommended Approach



Multi-Tiered Architecture



Data Warehousing & Business Intelligence Architecture



ETL – Extraction, Transformation and Load

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 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

The Design: Multidimensional

Dimensions

Facts

Org	Fiscal Qtr	Account	Amount	Encumbered	Balance
Dept Y	1	Travel	$1,000	$0	$60,000
Dept Y	1	Supplies	$500	$0	$59,500
Dept Y	2	Software	$10,000	$10,000	$49,500
Dept Y	2	Phones	$250	$0	$49,250
Dept Z	1	Travel	$8,000	$0	$100,000
Dept Z	1	Supplies	$100	$0	$99,900
Dept Z	2	Software	$80,000	$80,000	$19,900
Dept Z	2	Phones	$500	$0	$19,400

The Design: Dimensions

ACCOUNT_KEY
ACCOUNT_CODE
ACCOUNT_DESC
ACCOUNT_TYPE_CODE
ACCOUNT_TYPE_DESC
BUDGET_GROUP_CODE
BUDGET_GROUP_DESC

TIME_KEY
CALENDAR_PERIOD
CALENDAR_QUARTER
CALENDAR_DAY
CALENDAR_MONTH
CALENDAR_YEAR

GRANT_KEY
GRANT_CODE
GRANT_LONG_NAME
GRANT_SHORT_NAME
AGENCY_ID
AGENCY_NAME
GRANT_TYPE

ORG_KEY
ORGANIZATION_CODE
ORGANIZATION_DESC
EXECUTIVE_CODE
EXECUTIVE_DESC
SR_MANAGEMENT_CODE
SR_MANAGEMENT_DESC
DEPARTMENT_CODE
DEPARTMENT_DESC

The Design: Facts

ACCOUNT_KEY
ACCOUNT_CODE
ACCOUNT_DESC
ACCOUNT_TYPE_CODE
ACCOUNT_TYPE_DESC
BUDGET_GROUP_CODE
BUDGET_GROUP_DESC

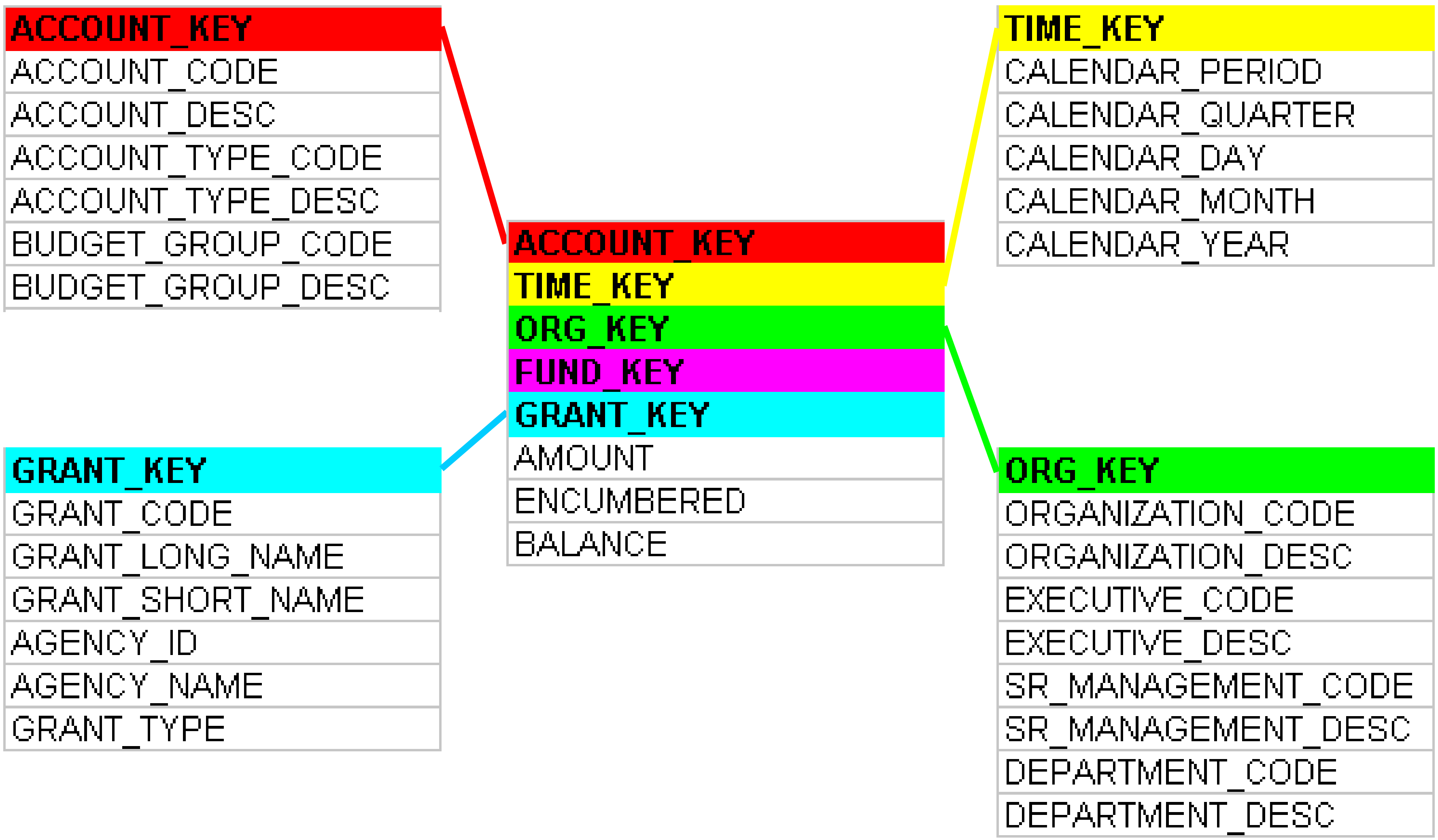
TIME_KEY
CALENDAR_PERIOD
CALENDAR_QUARTER
CALENDAR_DAY
CALENDAR_MONTH
CALENDAR_YEAR

ACCOUNT_KEY
TIME_KEY
ORG_KEY
FUND_KEY
GRANT_KEY
AMOUNT
ENCUMBERED
BALANCE

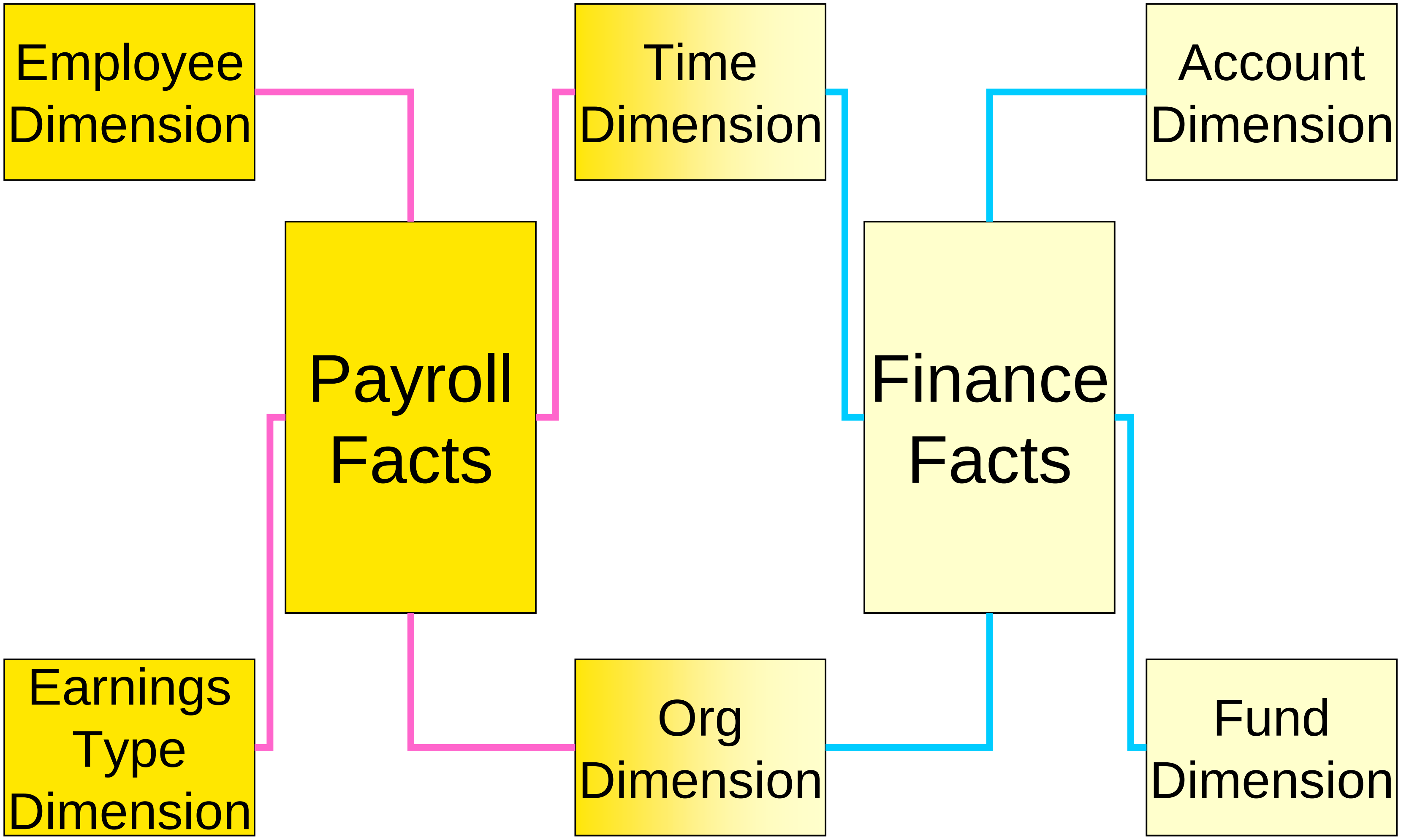
GRANT_KEY
GRANT_CODE
GRANT_LONG_NAME
GRANT_SHORT_NAME
AGENCY_ID
AGENCY_NAME
GRANT_TYPE

ORG_KEY
ORGANIZATION_CODE
ORGANIZATION_DESC
EXECUTIVE_CODE
EXECUTIVE_DESC
SR_MANAGEMENT_CODE
SR_MANAGEMENT_DESC
DEPARTMENT_CODE
DEPARTMENT_DESC

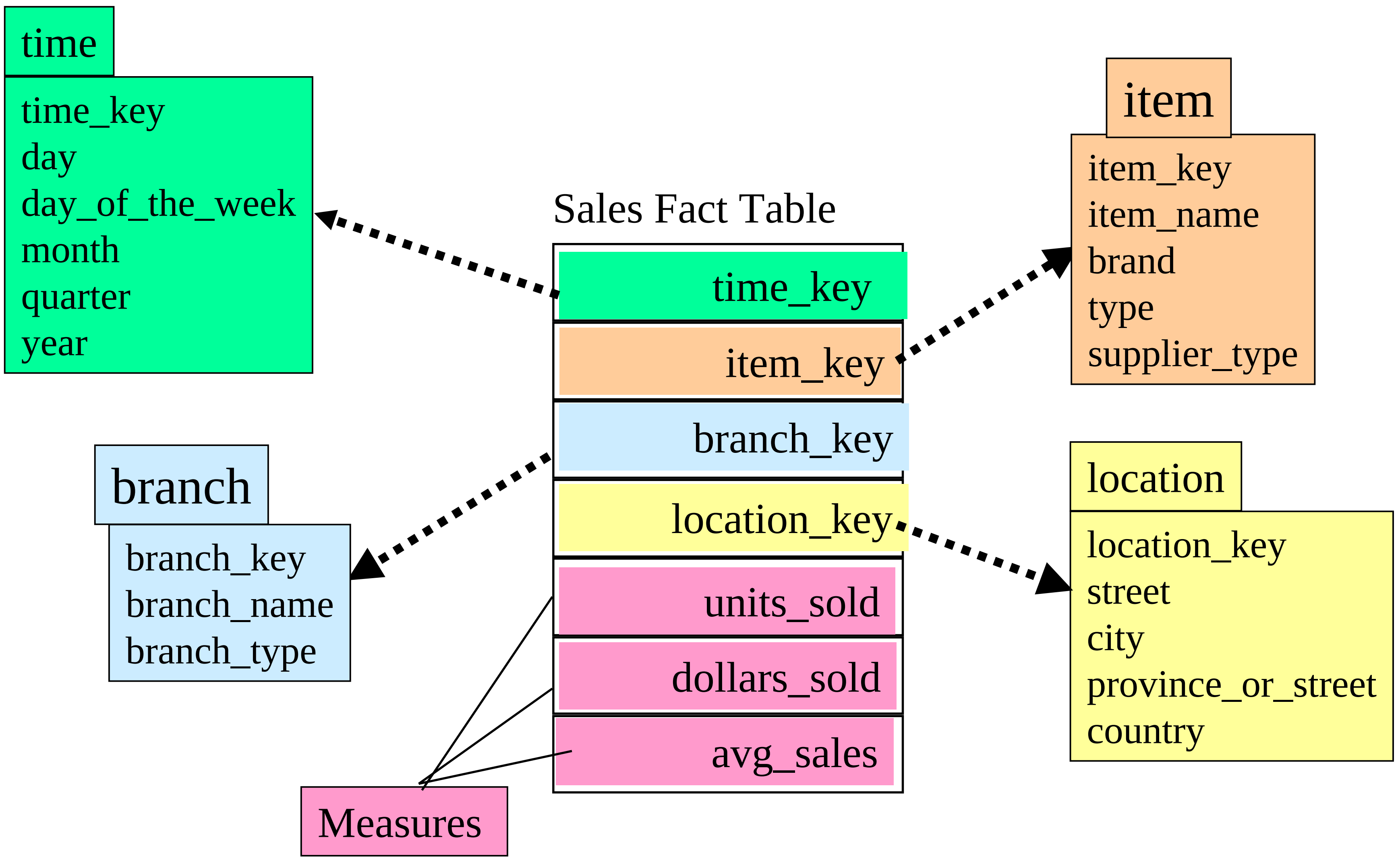
The Design: Star Schema



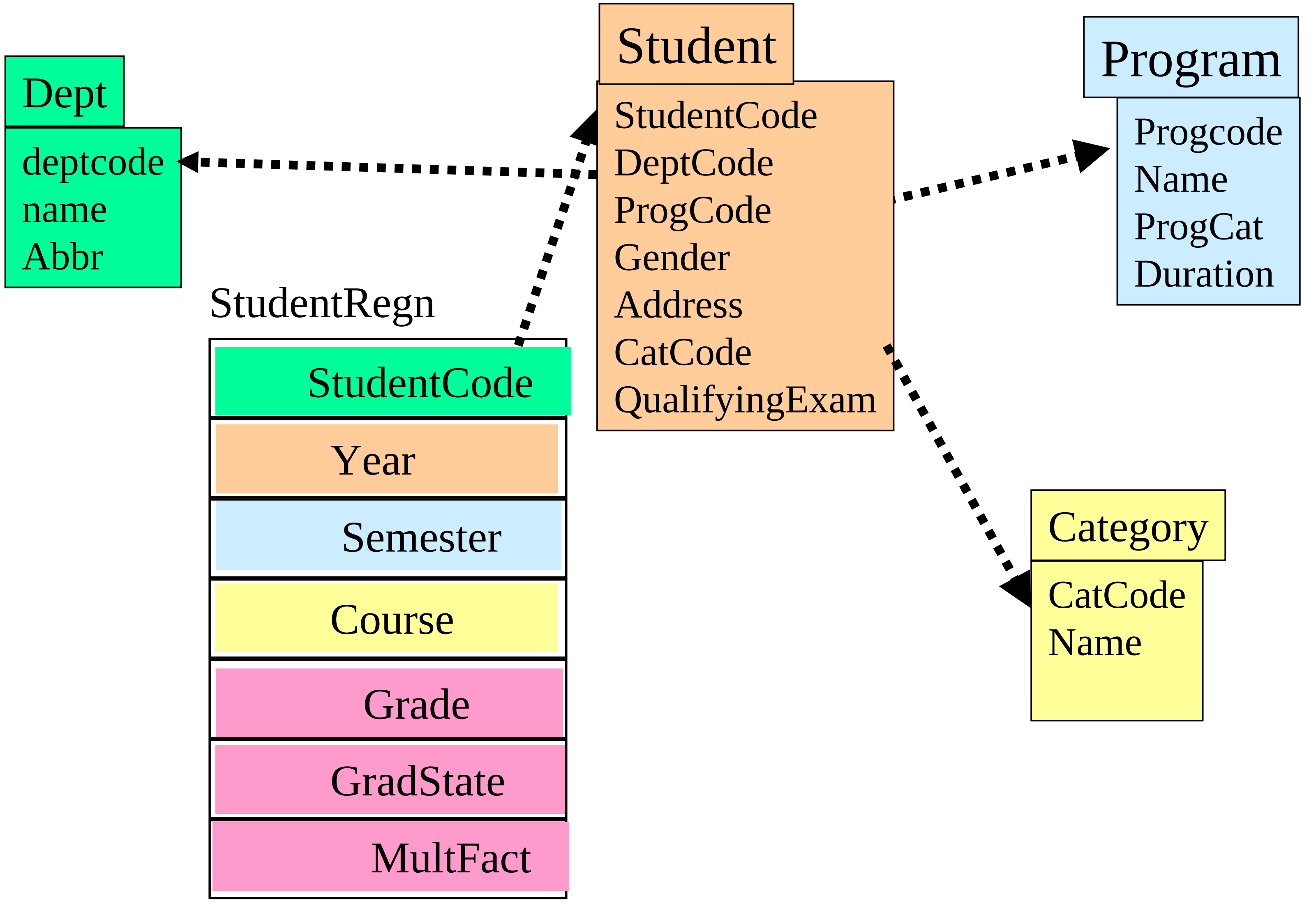
Example: Facts & Dimensions



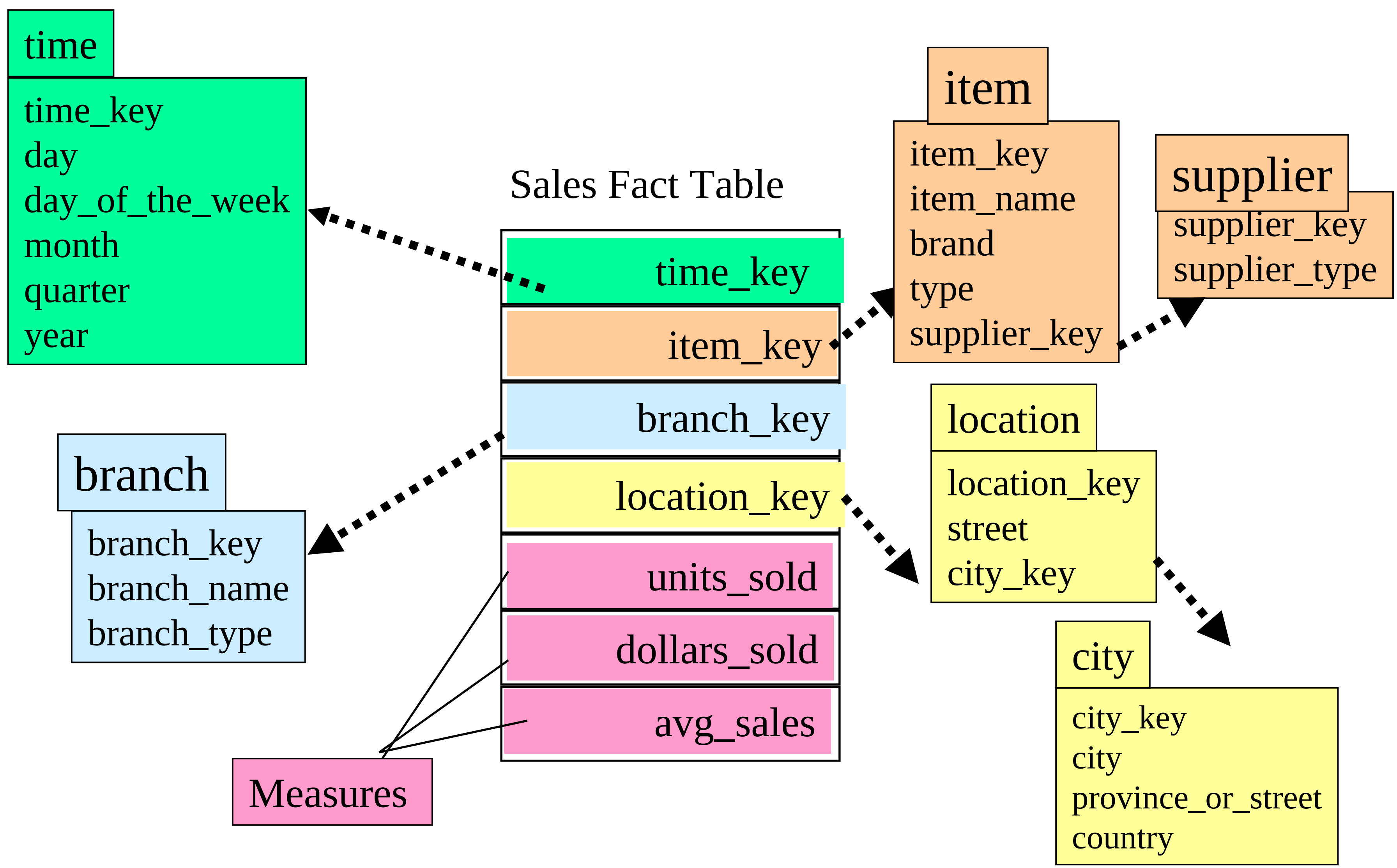
Another Example of Star Schema (cont'..)



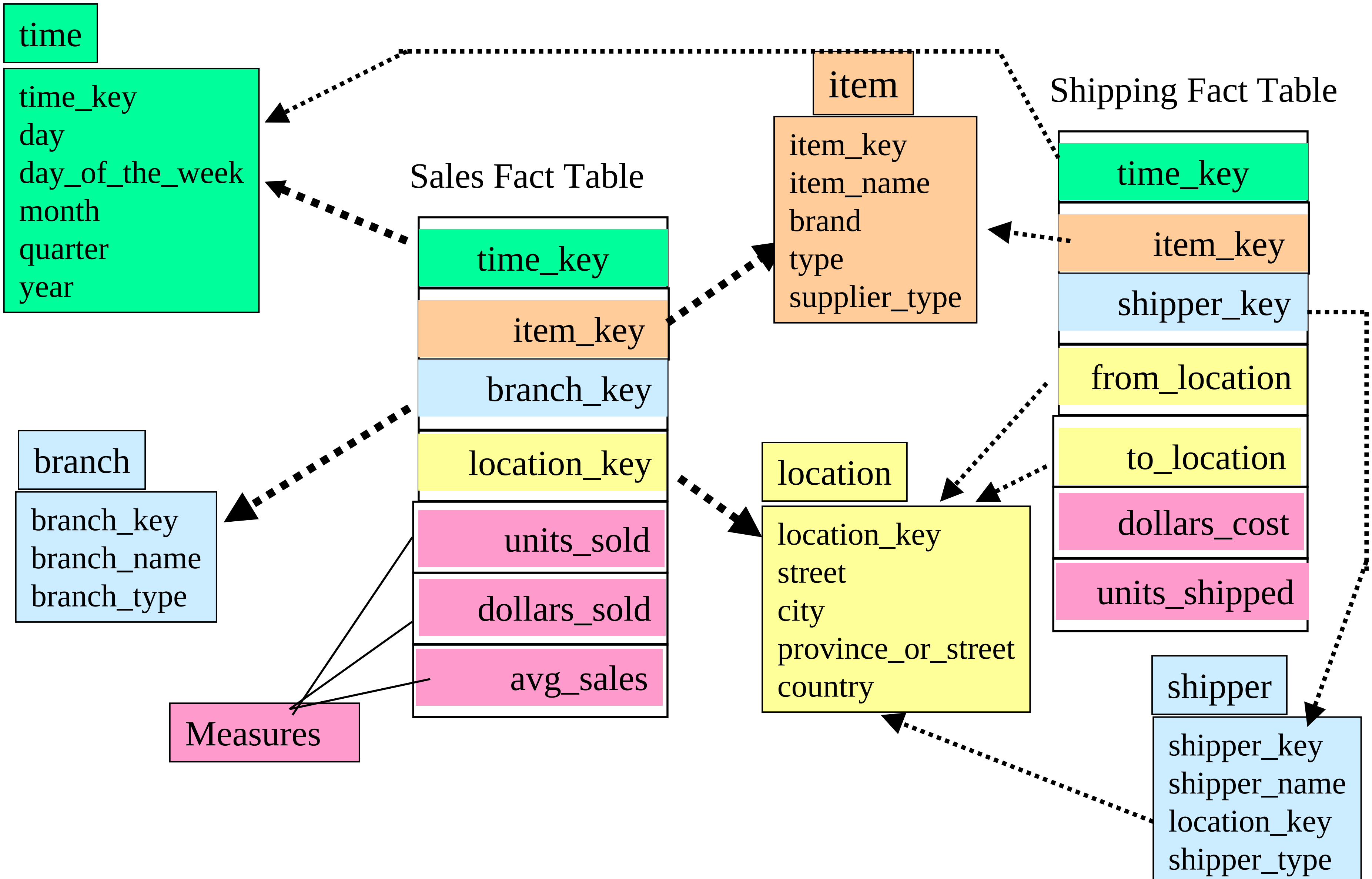
Example of Student database



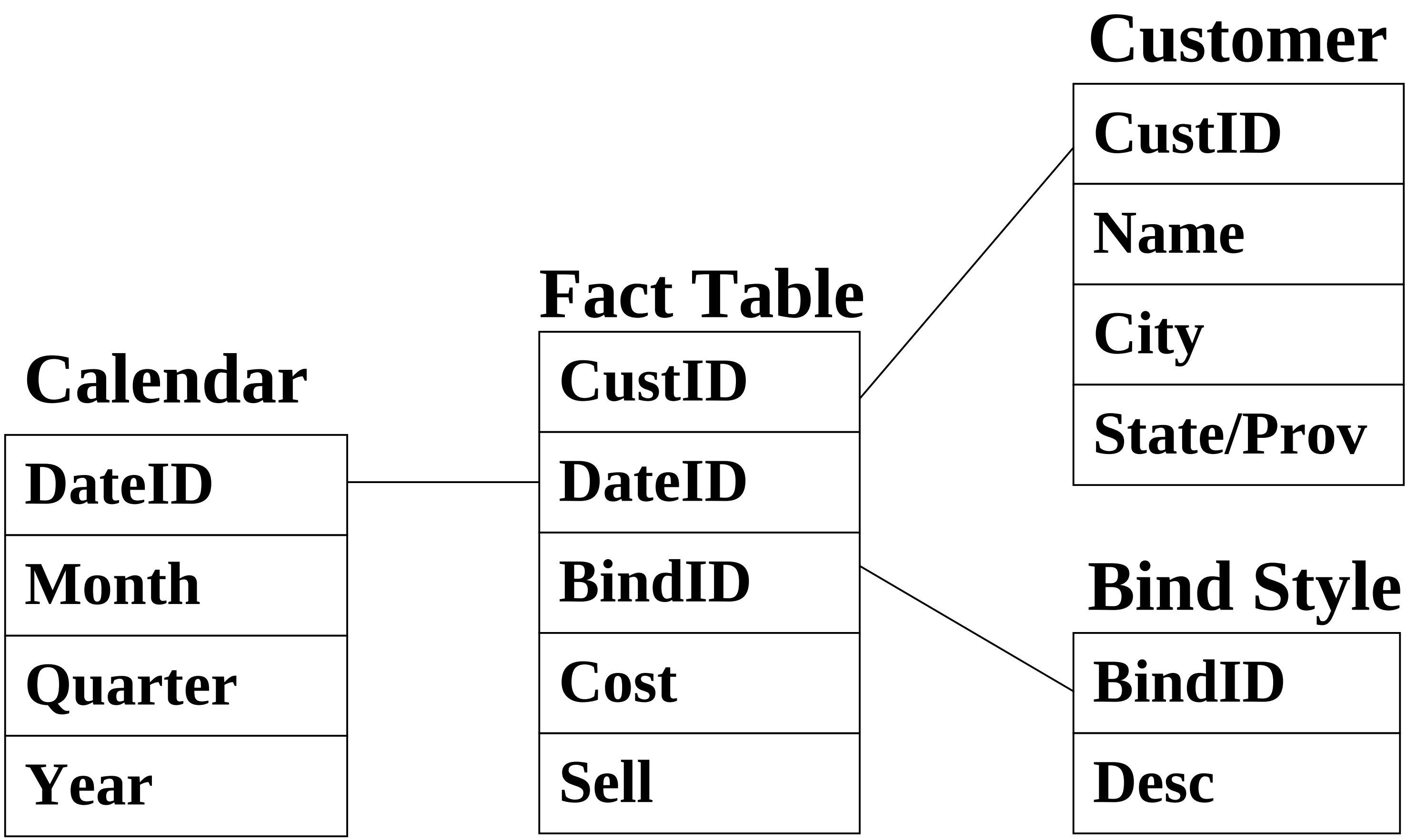
Example of Snowflake Schema



Example of Fact Constellation



Example Star Schema



Star Schema Viewed with Data

Customer

CustID	Name	City	State/Prov
00001	U of M	Ann Arbor	MI
00002	Smith & Co.	Toronto	Ont
⋮			

Calendar

DateID	Month	Quarter	Year
1/1/98	Jan	1	1998
1/2/98	Jan	1	1998
⋮			
12/31/00	Dec	4	2000

Fact Table

CustID	DateID	BindID	Cost	Sell
00002	12/31/00	PB	$500	$600
00222	1/1/99	HC	$1100	$1300
⋮				

Many Rows

Bind Style

BindID	Desc
PB	Paper Back
HC	Hard Cover

Conclusions

- A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process.

- A multi-dimensional model of a data warehouse
  - Star schema, snowflake schema, fact constellations
  - A data cube consists of dimensions & measures

- A data mart is a specialized system that brings together the data needed for a department or related applications.

SGG 4653
Advance Database System

Data Warehouse (Schema)

Contents

- Objectives of this topic:
  - To know and be able to define schema for a data warehouse
  - To understand multi-dimensional view of data in data warehousing
- Contents of this topic:
  - Data Mining Query Language, DMQL
  - Defining a Star Schema in DMQL
  - Defining a Snowflake Schema in DMQL
  - Defining a Fact Constellation Schema in DMQL
  - From Tables and Spreadsheets to Data Cubes

A Data Mining Query Language, DMQL: Language Primitives

- Cube Definition (Fact Table)

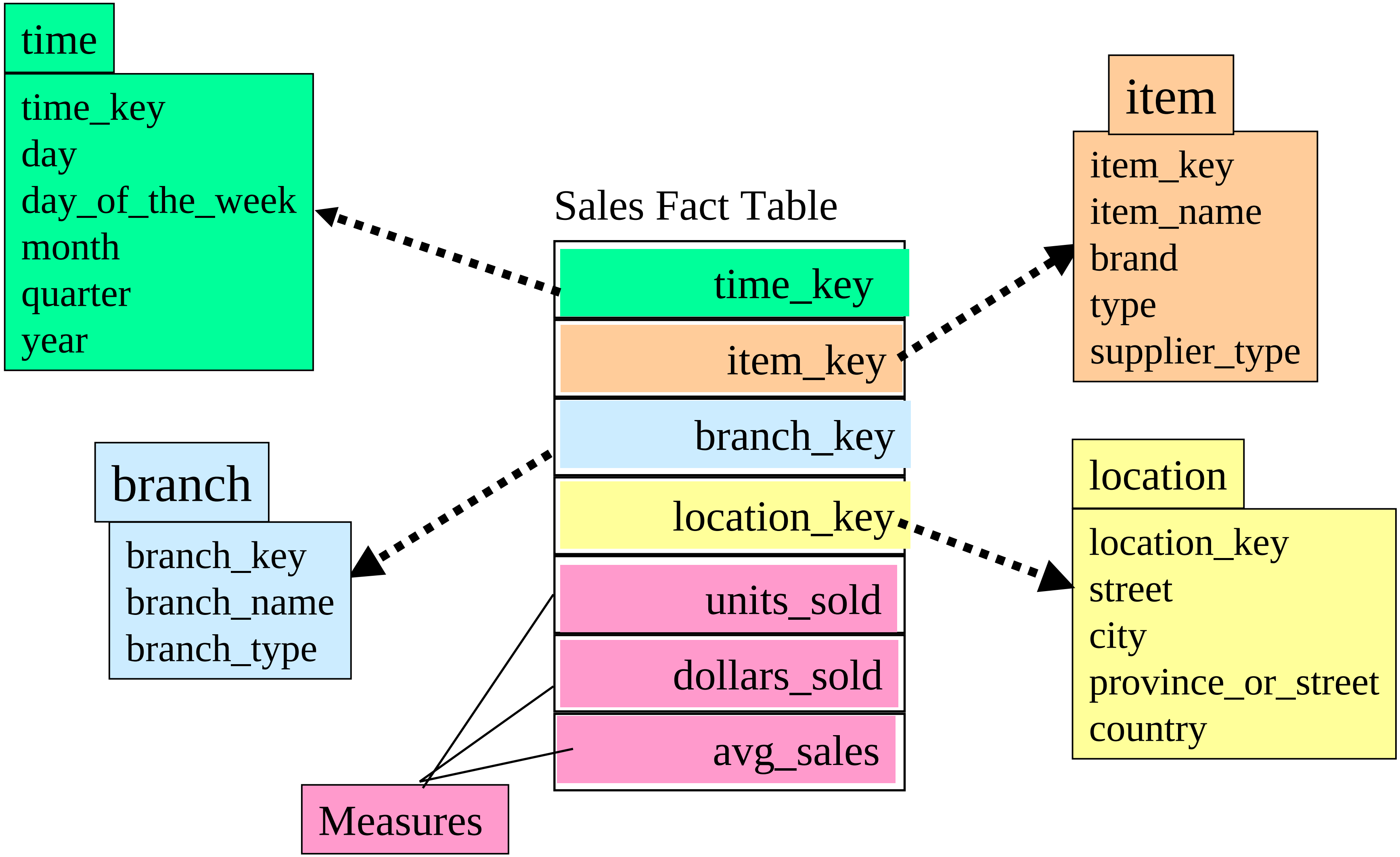
define cube <cube_name> [<dimension_list>]: <measure_list>

- 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>

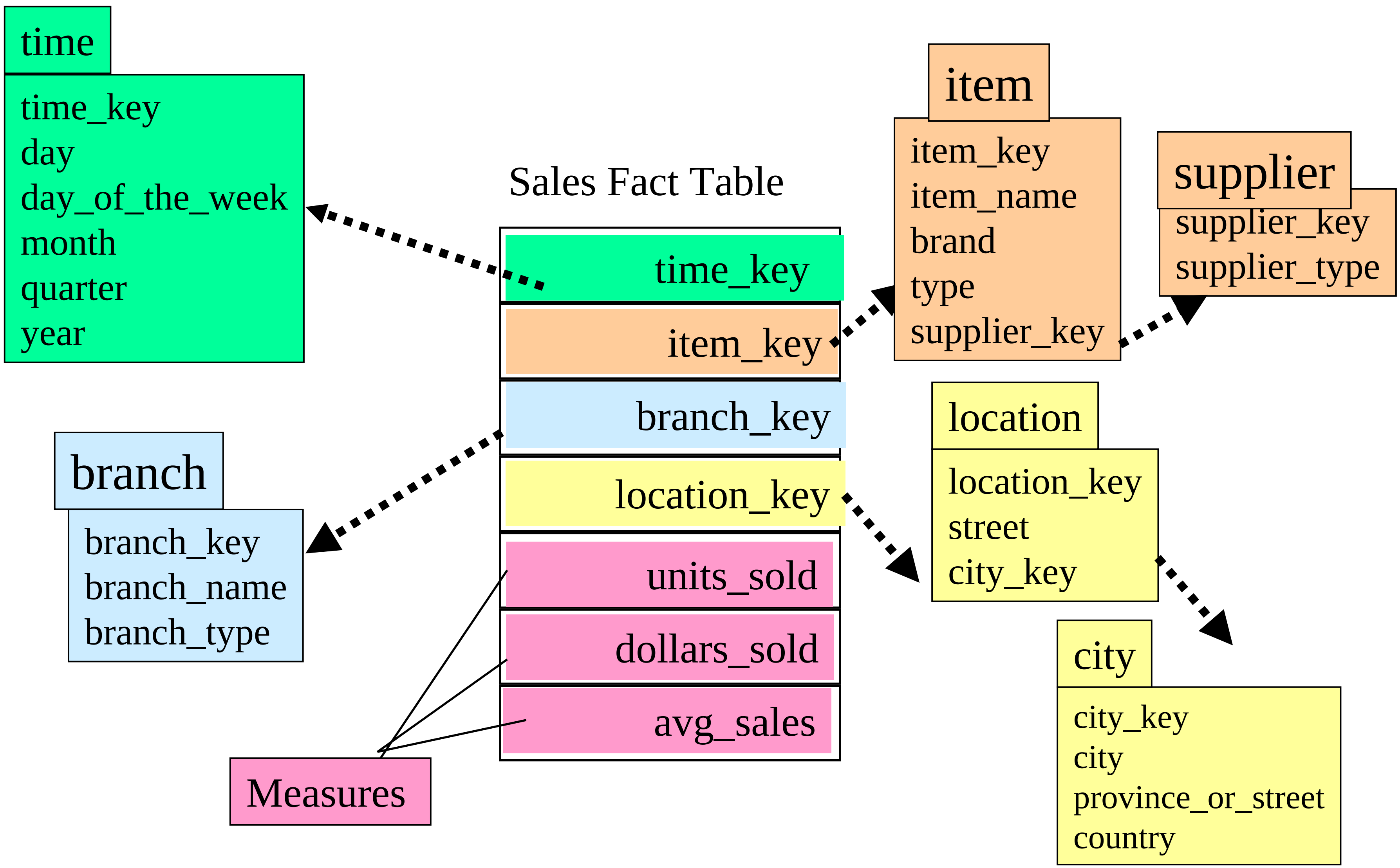
Example of Star Schema



Defining a 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)
```

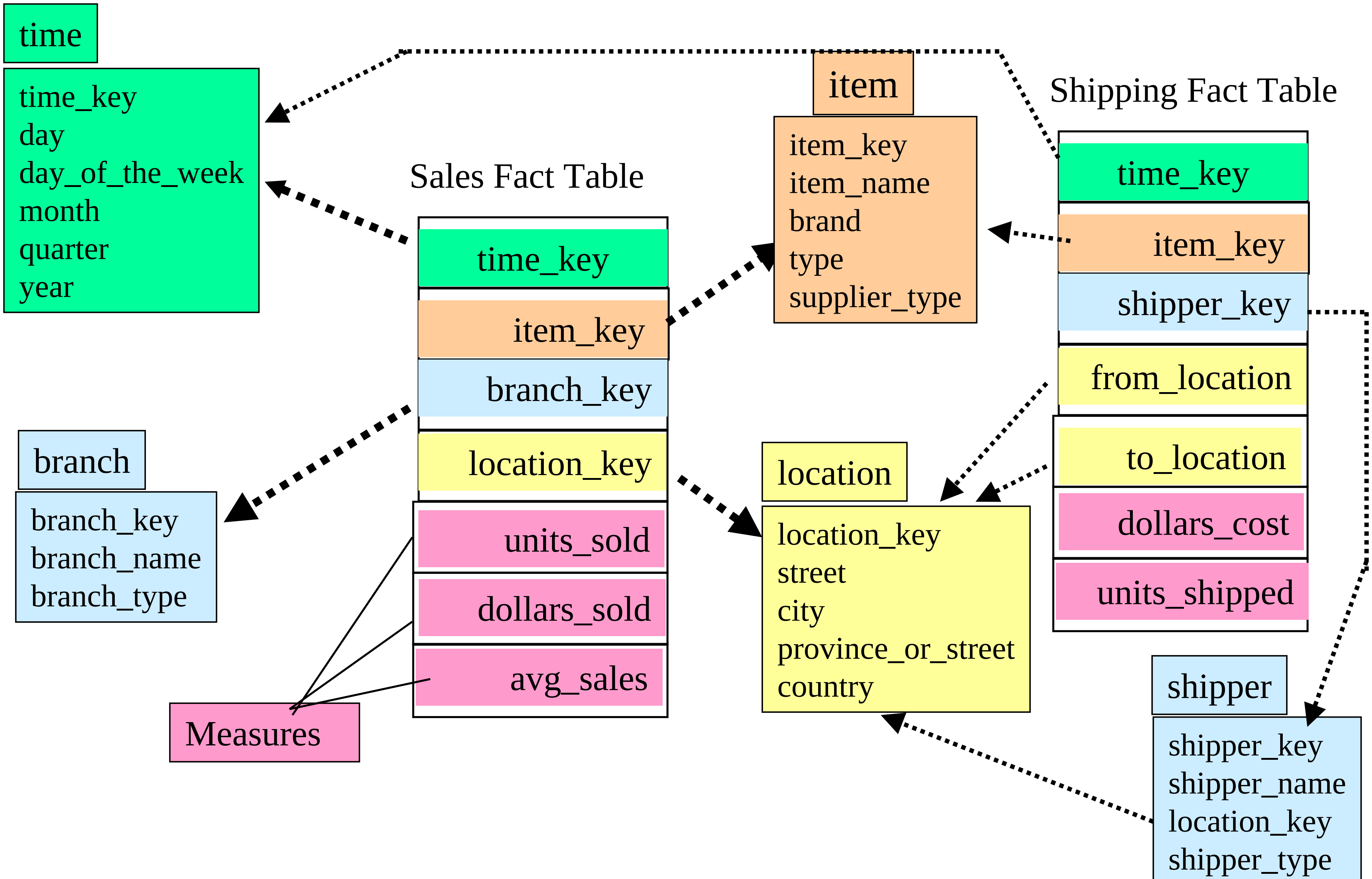
Example of Snowflake Schema



Defining a 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))
```

Example of Fact Constellation



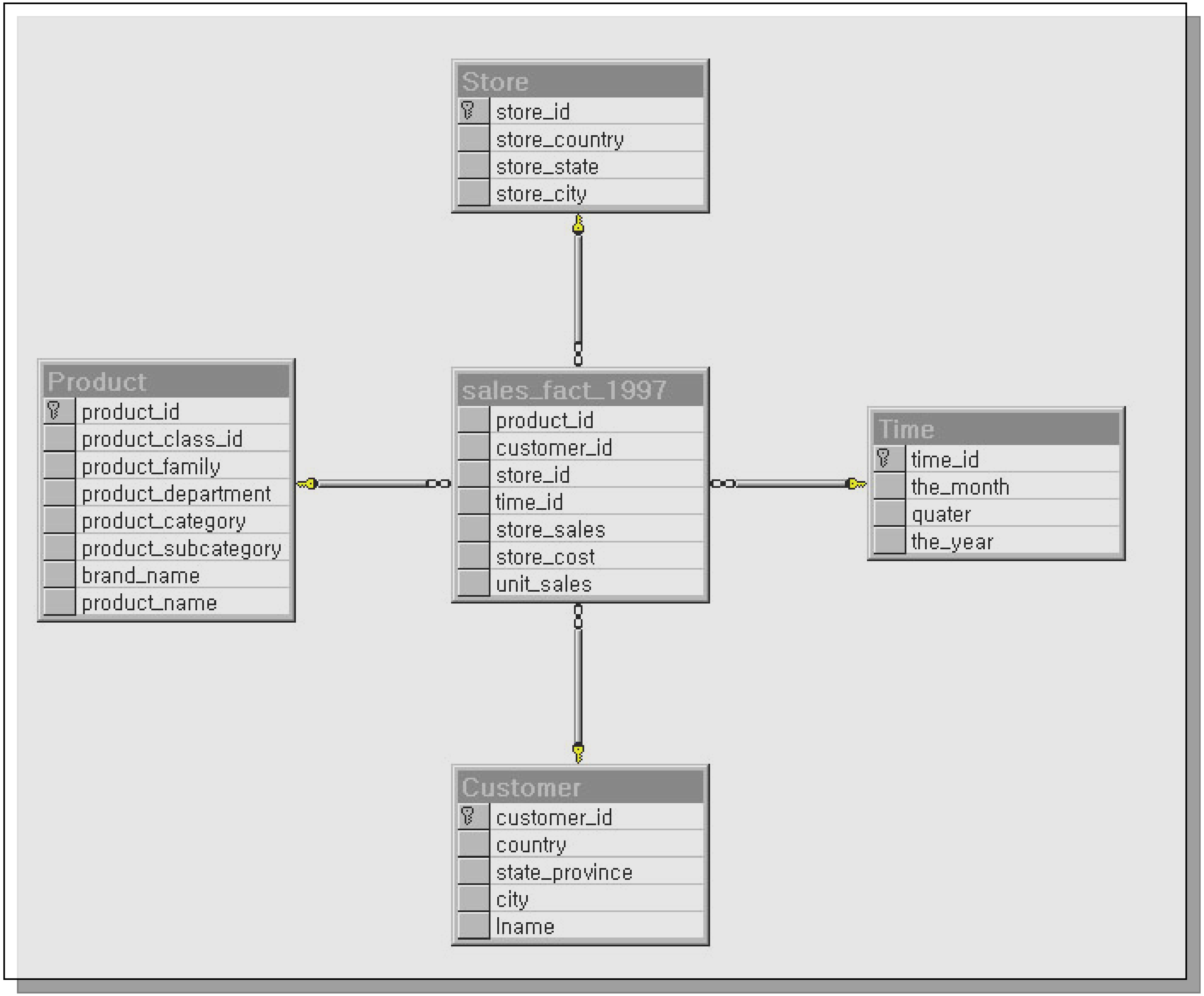
Defining a Fact Constellation in DMQL

```
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
```

Creating a Database

- Using CREATE DATABASE Options
  - **SIZE**
  - **MAXSIZE**
  - **FILEGROWTH**
- Setting Database Options
  - **Read-only: No Locking**
  - **Trunc. log on chkpt.: No Serious Recovery**
  - **SELECT INTO/Bulkcopy** : No Logging

Implementing of a Star Schema (example)



Implementing of a Star Schema (example)

- Creating Tables

Fact Table

```
CREATE table sales_fact_1997
( product_id int not null,
  customer_id int not null,
  store_id int not null,
  time_id int not null,
  store_sales float not null,
  store_cost float not null,
  unit_sales real not null )
```

Dimension Tables

```
CREATE table Customer
( customer_id int not null,
  country char(50) not null,
  state_province char(50) not null,
  city char(50) not null,
  lname char(100) not null,

  primary key (customer_id) )
```

```
CREATE table Product
( product_id int not null,
  product_class_id int not null,
  product_family char(50) not null,
  product_department char(50) not null,
  product_category char(50) not null,
  product_subcategory char(50) not null,
  brand_name char(255) not null,
  product_name char(255) not null,

  primary key (product_id) )
```

```
CREATE table Store
( store_id int not null,
  store_country char(50) not null,
  store_state char(50) not null,
  store_city char(50) not null,

  primary key (store_id) )
```

```
CREATE table Time
( time_id int not null,
  the_month char(15) not null,
  quater char(2) not null,
  the_year int not null,

 primary key (time_id) )
```

Implementing of a Star Schema (example)

- Define FOREIGN KEY Constraints

- **ALTER TABLE** sales_fact_1997
  ADD foreign key (customer_id) **references** Customer
- **ALTER TABLE** sales_fact_1997
  ADD foreign key (product_id) **references** Product
- **ALTER TABLE** sales_fact_1997
  ADD foreign key (time_id) **references** Time
- **ALTER TABLE** sales_fact_1997
  ADD foreign key (store_id) **references** store

•

create index fact
on sales_fact_1997 (product_id, customer_id, store_id, time_id)

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 subaggregate.

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

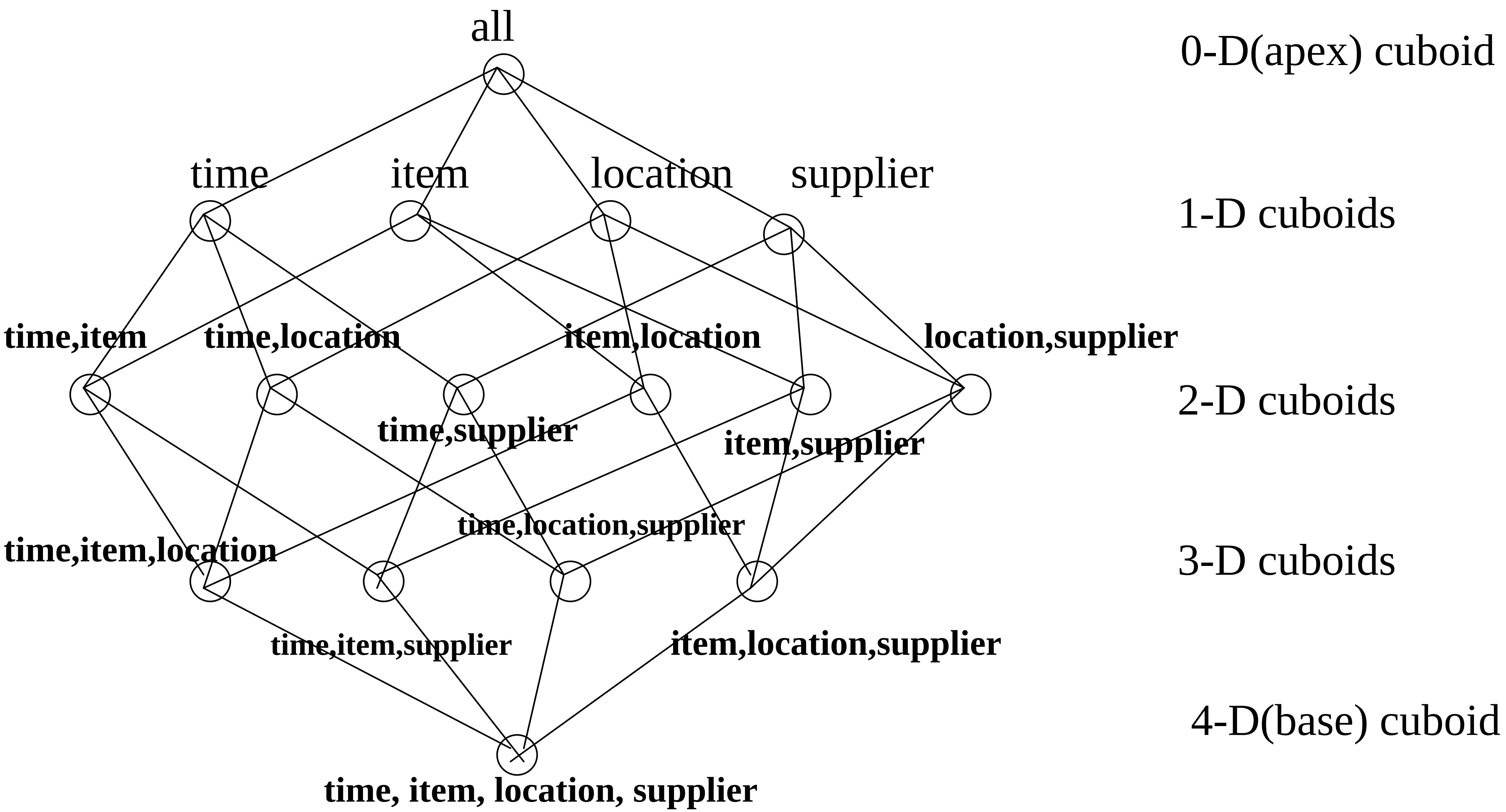
From Tables and Spreadsheets to Data Cubes

- A data warehouse is based on a multidimensional data model which views data in the form of a data cube

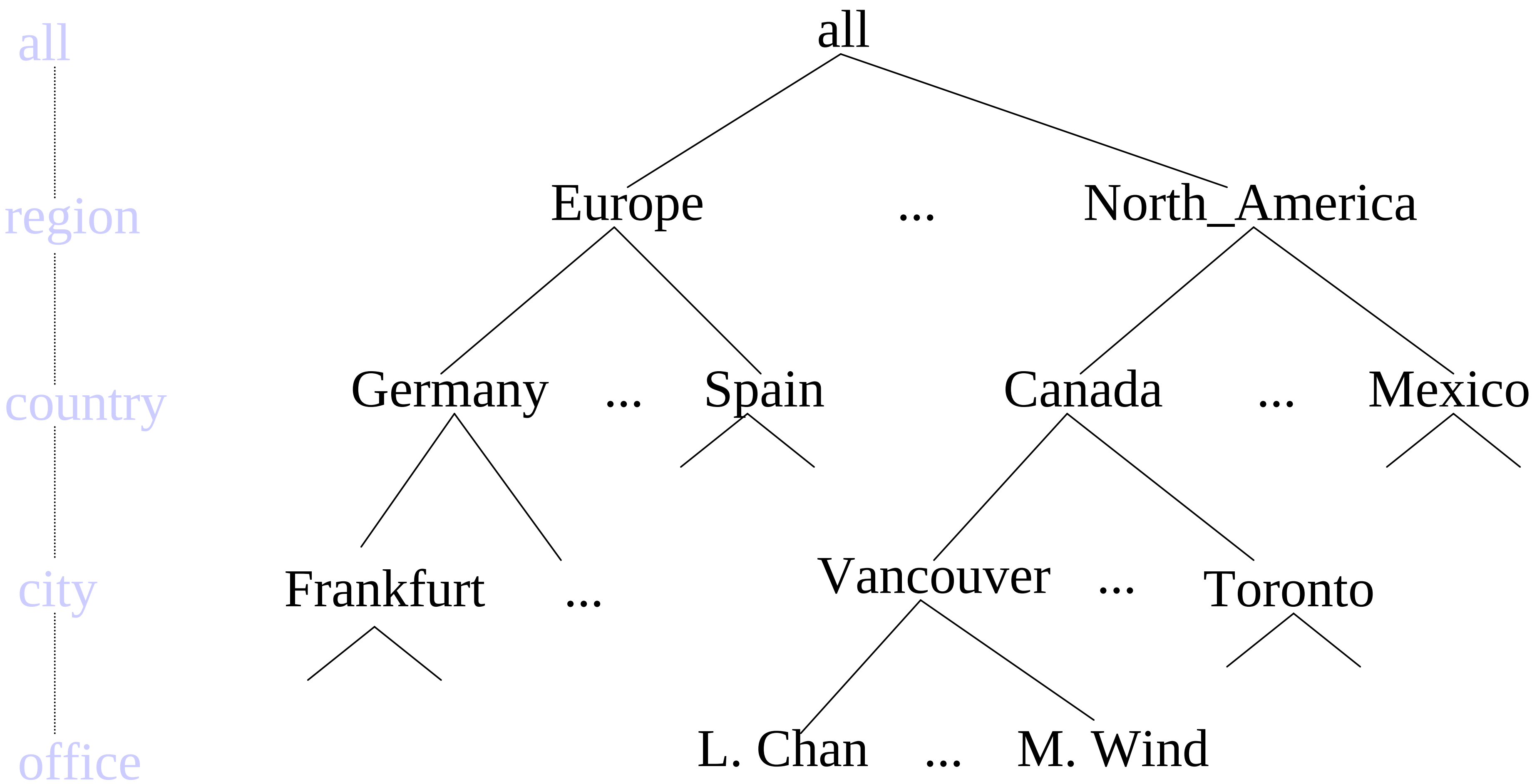
- A data cube, such as sales, allows data to be modeled and viewed in multiple dimensions
  - Dimension tables, such as item (item_name, brand, type), or time(day, week, month, quarter, year)
  - Fact table contains measures (such as dollars_sold) and keys to each of the related dimension tables

- 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.

Cube: A Lattice of Cuboids

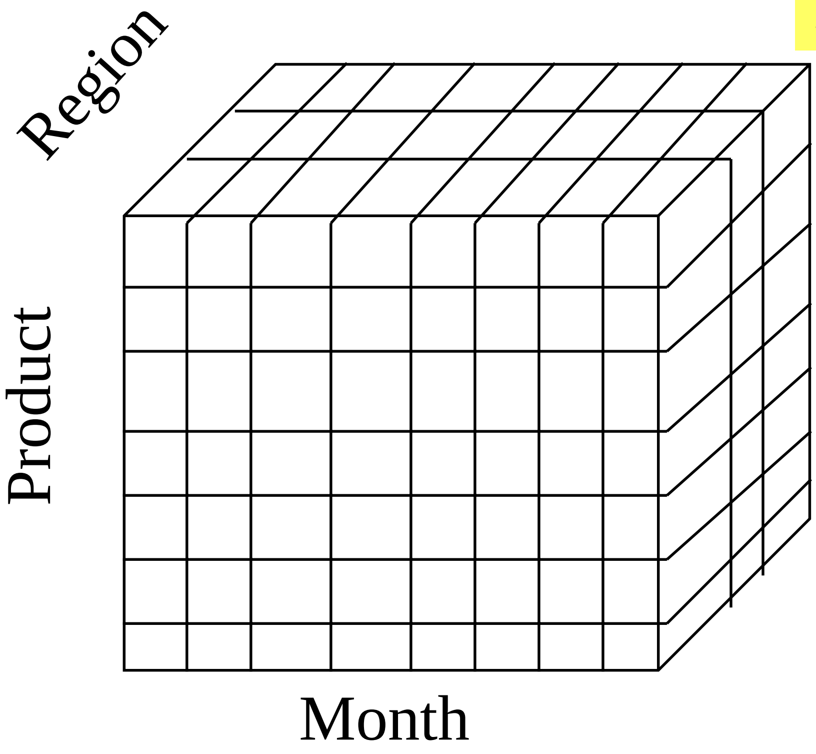


A Concept Hierarchy: Dimension (location)

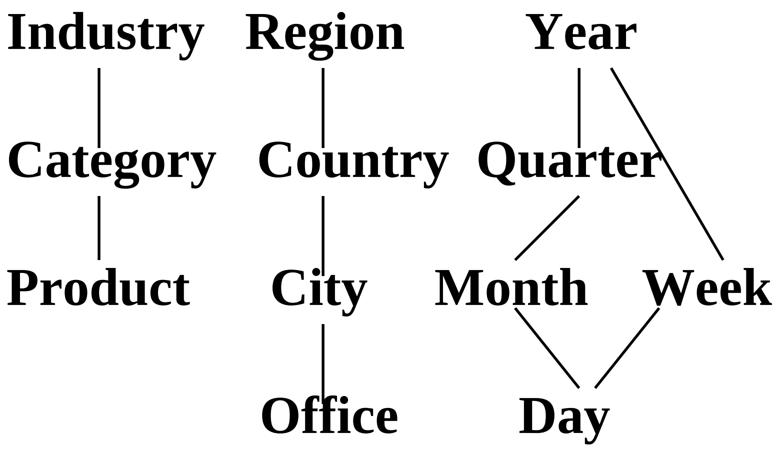


Multidimensional Data

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



Dimensions: Product, Location, Time
Hierarchical summarization paths



Cube

Fact table view:

sale	prodId	storeId	amt
	p1	c1	12
	p2	c1	11
	p1	c3	50
	p2	c2	8

Multi-dimensional cube:

	c1	c2	c3
p1	12		50
p2	11	8	

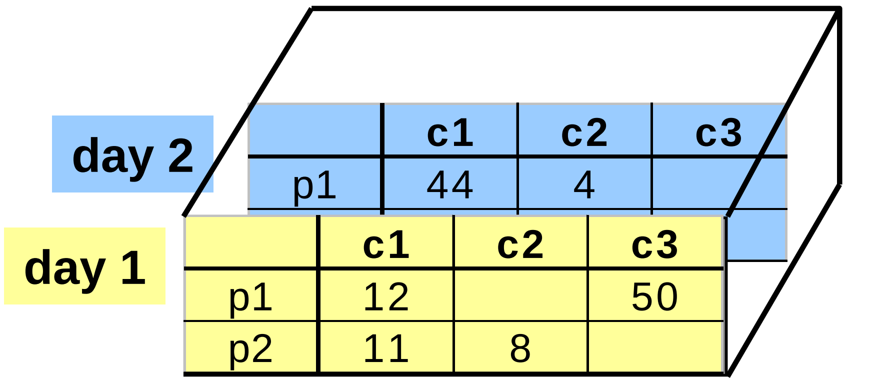
dimensions = 2

3-D Cube

Fact table view:

sale	prodId	storeId	date	amt
	p1	c1	1	12
	p2	c1	1	11
	p1	c3	1	50
	p2	c2	1	8
	p1	c1	2	44
	p1	c2	2	4

Multi-dimensional cube:



dimensions = 3

Conclusions

- A data warehouse is based on a multidimensional data model which views data in the form of a data cube
- A data cube, such as sales, allows data to be modeled and viewed in multiple dimensions
  - Dimension tables, such as item (item_name, brand, type), or time(day, week, month, quarter, year)
  - Fact table contains measures (such as dollars_sold) and keys to each of the related dimension tables
- 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