Data Warehousing

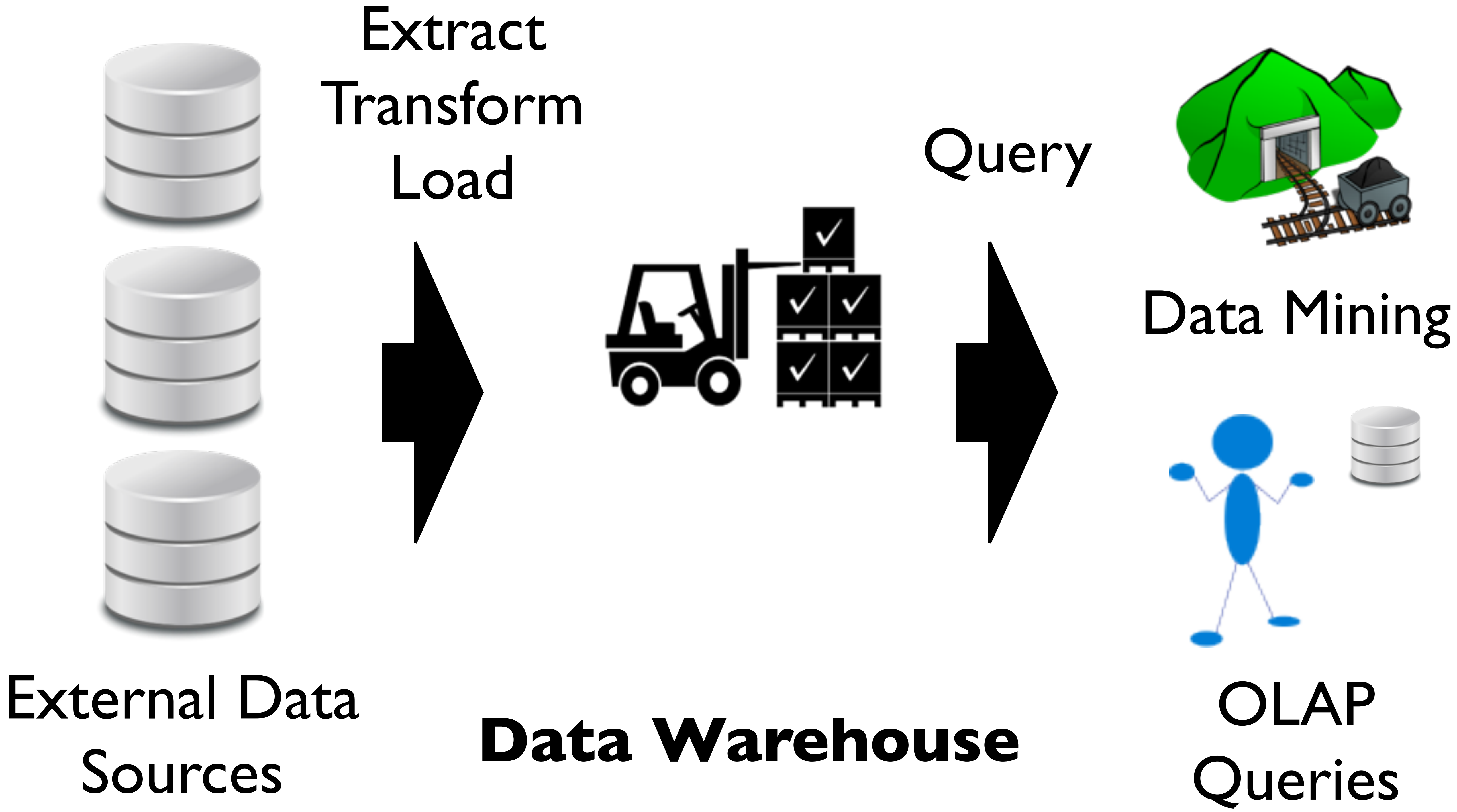
R&G Chapter 25

(slides adapted from content by J.Gehrke, J.Shanmugasundaram, and/or C.Koch)

Data Warehousing

- Big companies gather lots of data.
- This data can be exploited to…
  - …identify interesting patterns.
  - …correlate different data sources.
  - …support hypotheses/what if questions.

Data Warehousing



Extract
Transform
Load

Query

Data Mining

External Data Sources

Data Warehouse

OLAP Queries

image credit: openclipart.org

A Data Warehouse

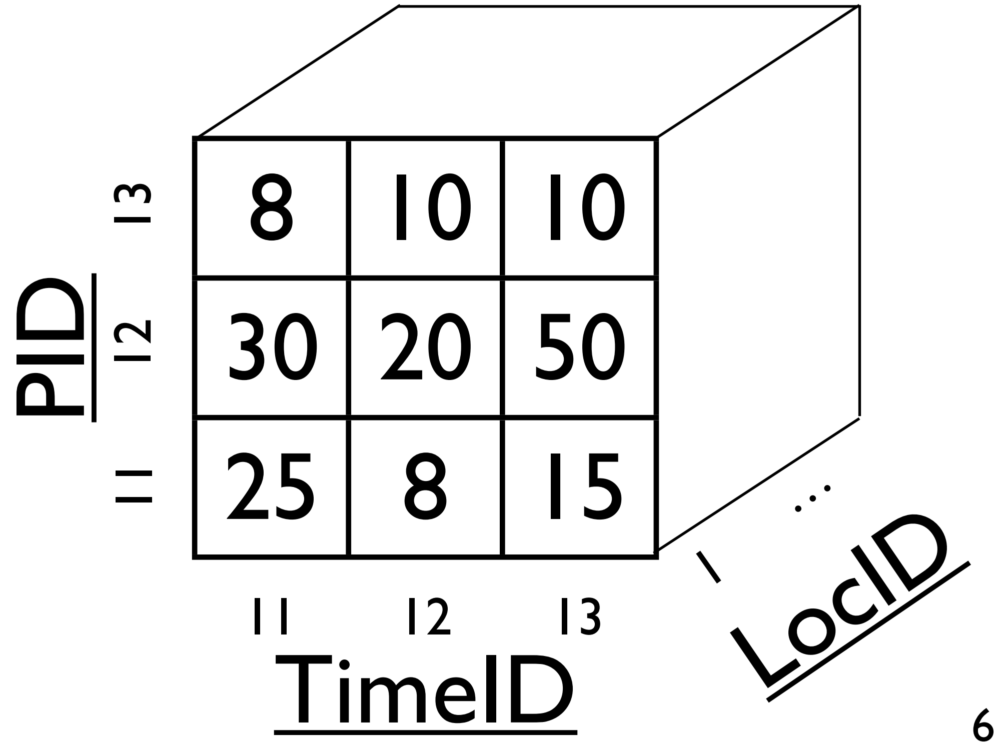
- Incorporates data spanning long time periods.
  - Merges data from many sources.
- Varies widely in scale: GB to TB to PB
- OLAP-centric workload:
  - Interactive query response times desirable.
  - Ad-hoc updates rare.

Warehousing Challenges

- **Semantic Integration**: Data coming from different sources has different schemas, identifiers for the same entities.
- **Heterogeneous Sources**: Data arrives/is stored in in a variety of different formats.
- **Load/Refresh**: Must load/organize data, and keep it up to date with the outside world.
- **Metadata Management**: Increasingly important to track provenance of source data.

Multidimensional Data Model (MOLAP)

- A Data Warehouse stores:
  - A collection of numeric measures…
  - … that depend on a set of dimensions.
- For example:
  - Sales by Product, Location, Time



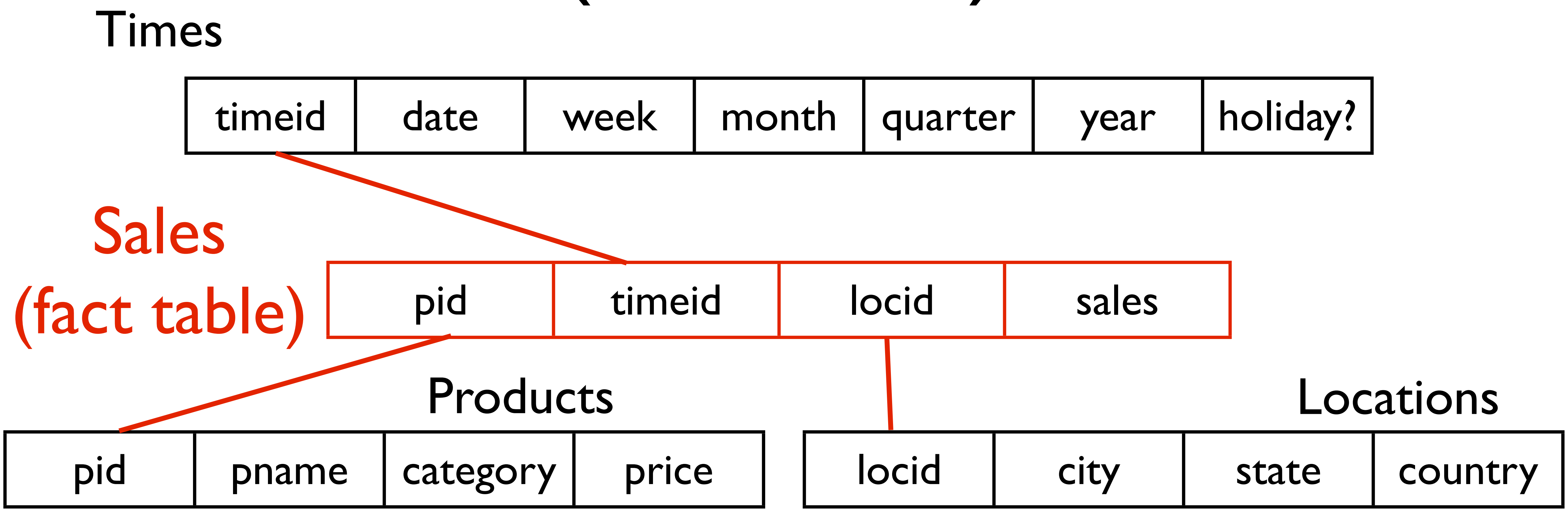
13	8	10	10
12	30	20	50
11	25	8	15
	11	12	13

PID	TimeID	LocID	Sales
11	1	1	25
11	2	1	8
11	3	1	15
12	1	1	30
12	2	1	20
12	3	1	50
13	1	1	8
13	2	1	10
13	3	1	10
11	1	2	35
…			

Relational Data Model (ROLAP)

- Central fact table relates multiple dimensions together.
    - Extremely large (nearly all data here)
- Each dimension can have additional attributes stored in a dimension table.
    - E.g. Products(pid, pname, category, price)

Relational Data Model (ROLAP)



Times

timeid	date	week	month	quarter	year	holiday?

Sales (fact table)

pid	timeid	locid	sales

Products

pid	pname	category	price

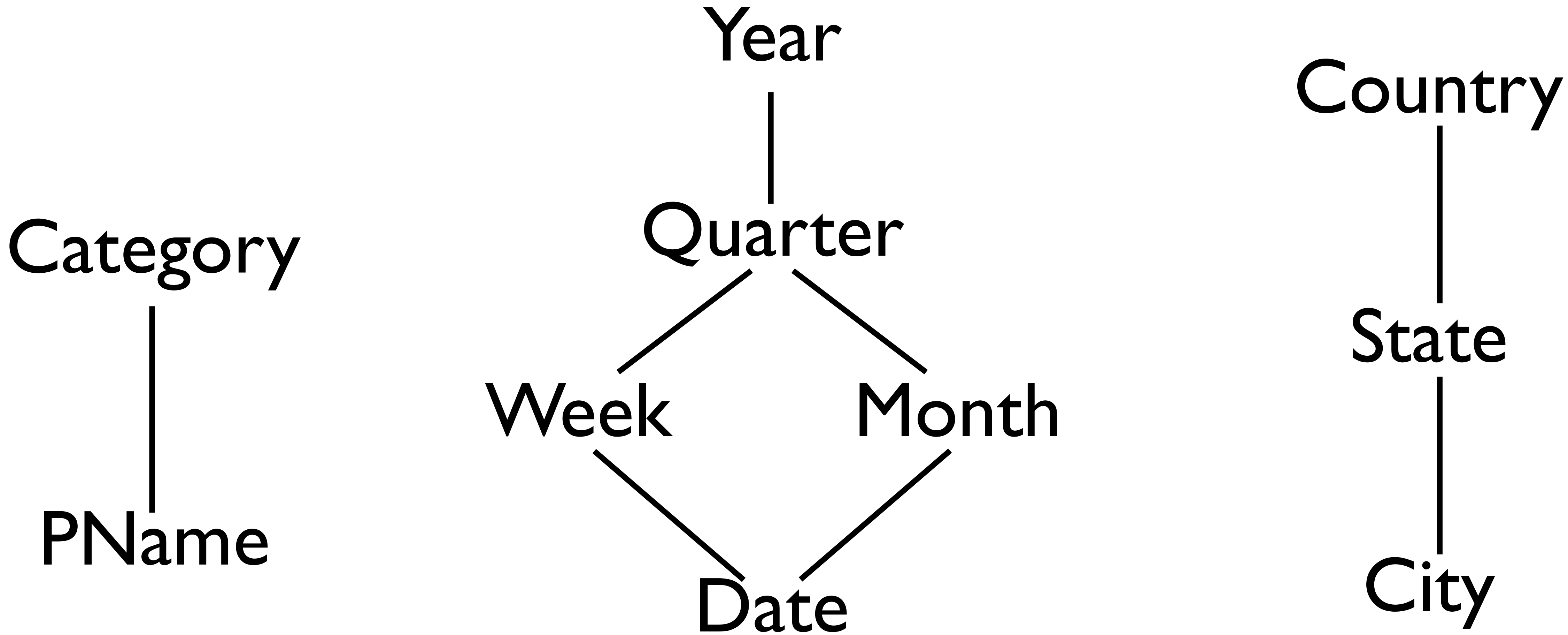
Locations

locid	city	state	country

- Dimension Tables stored Un-Normalized
    - Updates/Inserts; Unlikely to cause issues
- Common schema in OLAP: The <u>Star Schema</u>

Relational Data Model (ROLAP)

Dimension tables establish hierarchies of data



OLAP Queries

- Influenced by SQL and Spreadsheets
- Most Common Operation: Aggregate over one or more dimensions.
    - Find total sales
    - Find total sales for each city, for each state.
    - Find top five products by sales

OLAP Queries

- **Roll-Up**: Move to a coarser grained view of the data (more values aggregated)
    - "Moving up" in one dimension's hierarchy
- **Drill-Down**: Move to a finer grained view of the data (more aggregate values shown)
    - "Moving down" in one dimension's hierarchy

OLAP Queries

- **Pivoting**: Aggregate over selected dimensions to get a cross-tabulation.
- **Slice and Dice**: Equality or Range Selections on one or more dimensions.

	NY	WA	**Total**
2012	63	81	144
2011	38	107	145
2010	75	35	110
Total	176	223	339

OLAP Queries vs SQL Queries

OLAP Queries vs SQL Queries

- Cross-tabulations can be obtained from SQL Queries. How?

OLAP Queries vs SQL Queries

- Cross-tabulations can be obtained from SQL Queries. How?
- Generalizing, how many different group-by queries can we create over a fact table with k dimensions?

OLAP Queries vs SQL Queries

- Cross-tabulations can be obtained from SQL Queries. How?
- Generalizing, how many different group-by queries can we create over a fact table with k dimensions?
- The CUBE operator creates all of these.

OLAP Queries vs SQL Queries

- Cross-tabulations can be obtained from SQL Queries. How?
- Generalizing, how many different group-by queries can we create over a fact table with k dimensions?
- The CUBE operator creates all of these.
    - Prebuilds cube for faster OLAP queries.

The CUBE Operator

- CUBE pid, locid, timeid BY SUM Sales
  - Contains results of all 8 possible group-by queries.
  - How would this be represented in SQL?
- This is an expensive operation, how might it be implemented efficiently?

Bitmap Indexes

- Slice & Dice requires Equality Tests.
- Testing for equality is expensive (pipeline stalls)
  - Can we aggregate using purely arithmetic operations?
- Store enumerations as bitmaps
  - One bit per possible value: Each row sets one bit.

Bitmap Indexes

Data				Bitmaps	
CustID	Name	Gender	Rating	Gender	Rating
112	Alice	F	3	01	00100
115	Bob	M	5	10	00001
119	Carol	F	5	01	00001
113	Dave	M	4	10	00010

```
SELECT SUM(Sales)
WHERE Gender = ‘M’
AND Rating = 5
```

```
Tot += Sales * (
    ((G & 0x2) >> 1)
   &((R & 0x1) >> 0)
)
```

Join Index

- Consider the Join of Sales, Products, Times, Locs with the predicate Country = ‘USA’
  - A join index helps speed this process up.
  - < s, p, t, l > in the index if SaleID s (respectively p, t, l) matches the predicate.
- Problem: Too many, too big join indexes.
  - Solution: Index by value and SaleID (fact table rowid)
  - To match multiple predicates, intersect columns.

Sequences

- Trend analysis is difficult (in SQL 92)
  - Find the % change in monthly sales
  - Find the daily top-5 product by sales in the last week
  - Find the trailing n-day moving average of sales.
- The first two examples are hard to express.
  - The third can not be expressed if n is a parameter
- SQL 99's WINDOW clause analyzes sequences.

WINDOW

- Define a sequence (by sorting a relation)
- Generate all subsequences of size N.
- Compute aggregates for each subsequence.

WINDOW

```
SELECT L.state, T.month,
       AVG(S.sales) OVER W as movavg
FROM   Sales S, Times T, Locations L
WHERE  S.timeid = T.timeid
  AND  S.locid = L.locid
WINDOW W AS (
   PARTITION BY L.state
   ORDER BY T.month
   RANGE BETWEEN INTERVAL ‘1’ MONTH PRECEDING
         AND INTERVAL ‘1’ MONTH FOLLOWING
)
```