THE UNIVERSITY OF QUEENSLAND AUSTRALIA | CREATE CHANGE

Advanced Database Systems (INFS3200)

Lecture 5: Data Warehouses

Lecturer: Prof Shazia Sadiq

School of Electrical Engineering and Computer Science (EECS)

Faculty of Engineering, Architecture and Information Technology

The University of Queensland

Topics

Why do we need Data Warehouses

Building a Data Warehouse

Multi-dimensional Data Models

Data Warehouse Design

OLAP Queries

CREATE VIEW

Consider schema

- EMPLOYEE [SSN, fname, lname, address, dept, salary]
- PROJECT [Pno, pname, budget, manager]
- WORKS-ON [SSN, Pno, hours]

CREATE VIEW

```
CREATE VIEW WORK-HOURS AS
SELECT lname, SUM(hours) AS TH
FROM EMPLOYEE E, PROJECT P, WORKS-ON W
WHERE E.SSN = W.SSN AND
          P.Pno = W.Pno
GROUP BY lname
```

List all employees who work for more than 50 hours across all projects

```
SELECT lname, TH
FROM WORK-HOURS
WHERE TH > 50
```

Why do we need Data Warehouses?

Traditional database applications consist of **both updates and queries**

- While, some queries are large scale aggregation reports which can take long time to generate on-the-fly

Database updates and queries must lock data resources

- Large scale aggregation reports lock many resources for a long time

If high frequency of database updates coincides with high frequency of reports, there is **competition** for computing resources

- For example, student enrolment transactions at beginning of semester coincide with high report demand for checking if room sizes, tutor allocations etc are adequate

Data Warehouse is useful

Organizations are analysing current and historical data to identify useful patterns and support business strategies

Emphasis is on complex, interactive, exploratory analysis of very large datasets created by integrating data from across all parts of an enterprise

Resource competition solved by making periodic replicas of data from operational data into separate system for analytics

- Data snapshots are acceptable
- Pre-processing for common aggregations are desirable
- Efficient support for common analytics operations

OLTP (Online Transaction Processing) vs. **OLAP** (Online Analytical Processing)

OLTP system is a database system used to record current **Update, Insertion** and **Deletion** transactional operations.

- Queries are simpler and short
- Time-critical in processing, and requires less space

OLAP database **stores historical data** that has been collected from OLTP databases

- view different **summaries** of **multi-dimensional** data
- extract information from **a large database**
- analyse data for **decision making**

OLTP vs. OLAP (cont.)

OLTP is an online **transaction** system whereas, OLAP is an online **data retrieval and analysis** system.

Transactional data is the source of OLTP, whereas different OLTP databases are the source of OLAP.

OLTP's main operations are **insert, update** and **delete** whereas, OLAP's main operation is to **extract multidimensional data for analysis**.

OLTP has **short but frequent** transactions whereas, OLAP has **long and less frequent** transaction.

Processing time for the OLAP's transaction is more as compared to OLTP.

OLAPs queries are more **complex** with respect OLTPs.

The tables in OLTP database must be **normalized** (**3NF**) whereas, the tables in OLAP database **may not be normalized**.

As OLTPs frequently executes transactions in database, in case any transaction fails in middle and hence it must take care of data integrity. While in OLAP the transaction is less frequent hence, it does not bother much about data integrity.

https://techdifferences.com/difference-between-oltp-and-olap.html#:~:text=OLAP

A Data Warehouse has

Integrated data spanning long time periods, often augmented with summary information

Very large volume: several Terabytes (TB) common

Interactive response times expected for complex queries

Ad-hoc updates uncommon (***Write-once*** and ***Read-forever***)

Responding times: simple query: <1s, complex query: <3s, really complex query: <6s

Data Warehousing Environments

In Data Warehouse (DW), data is decoupled from its generation source

Information in DW is organized to be easily used for DSS applications (i.e., a variety of visualization charts)

- Database views are used to organize data for DW

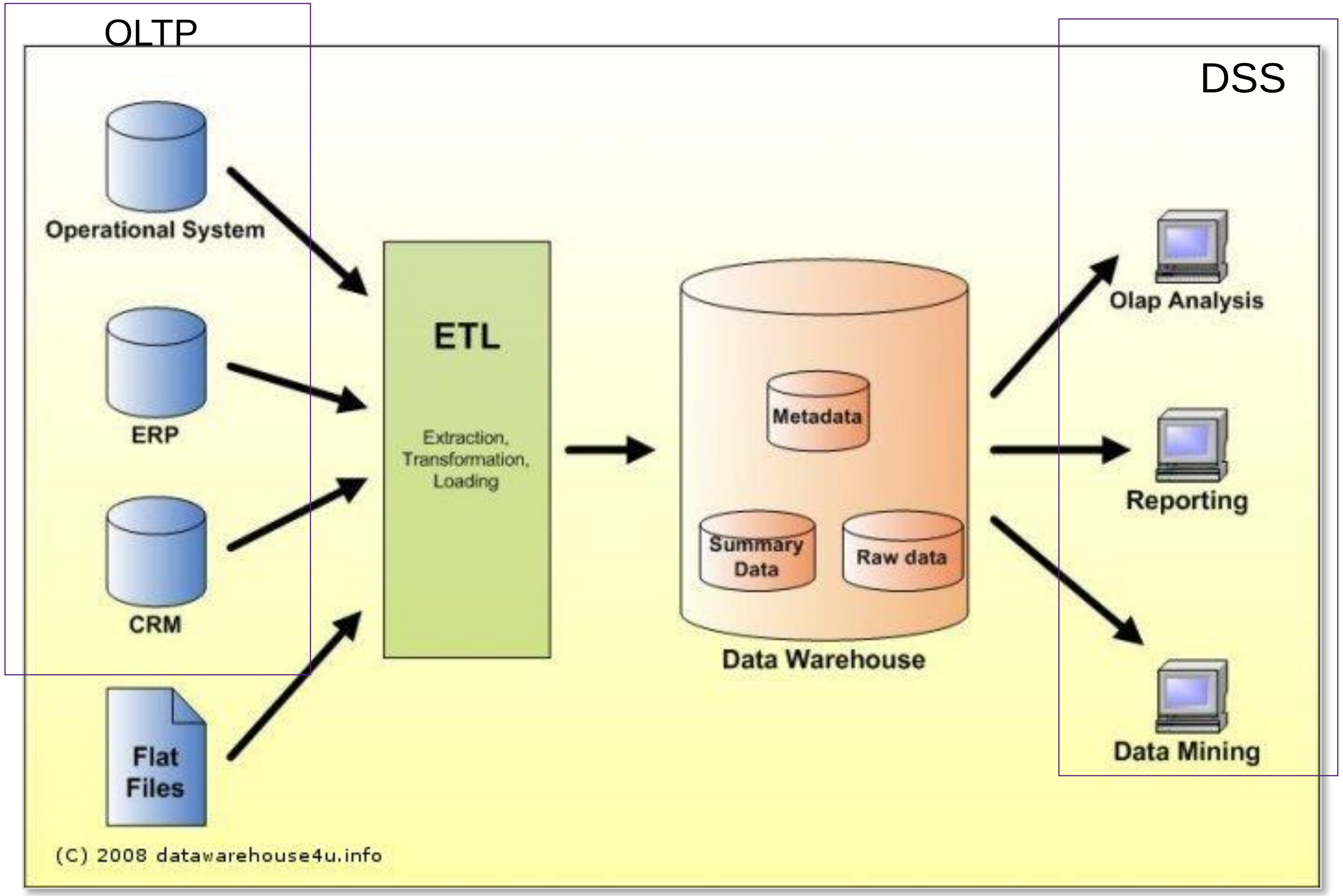
Information is available independently from the availability of the source

- The **views** are materialized

Information is structured and stored in order to optimize processing of DW queries

Only a small cooperation is required with the source to keep the warehouse in sync of time periods

Data Warehouse Overview



Name three differences between DBMS and Data Warehouse

Name three differences between Transaction Processing Databases and Data Warehouses

For insert, delete, update operations but DW is read only

For operational support but DW is primarily for decision support

Stores transactions but DW stores aggregated data

Building a Data Warehouse

1. The data must be **extracted** from multiple, heterogeneous sources (i.e., from OLTP databases)
2. The data must be **transformed** to fit into the data warehouse model where

- The data must be formatted for consistency of multiple sources
- The data must be cleaned to ensure validity
- The data must be fitted to the DW data model (pre-processed for summary data)

3. The data must be **loaded** into the DW

DBMS vs Data Warehousing

Sales (Summary Data)

Day	Product	Store	Sales ($)
9/2/2014	Milk	Toowong	3412
10/2/2014	Bread	Toowong	3445
9/2/2014	Milk	Kenmore	5440
10/2/2014	Bread	Kenmore	3067
…	…	…	…

- What is the total sale in each store?
- How about milk sold on Monday?
- Which item is the most popular one?

Aggregated result

Purchase (Operational Data)

Day	Product	Store	Qty	Price
9/2/2014	A2 milk	Toowong	1	3.3
9/2/2014	Grape green	Toowong	2	7.9
9/2/2014	Lindt choc	Kenmore	1	8.4
9/2/2014	Coles coke	Kenmore	2	3.2
…	…	…		…

- Change the Price of “A2 milk” to $4 each.
- Delete the “Grape green” sold on “9/2/2014” in “Toowong”

Data Warehousing Issues

Syntactic data integration

➢Must access data from a variety of source formats and repositories

Semantic data integration

➢When getting data from multiple sources, must eliminate mismatches, e.g., different currencies

Load, refresh and purge

➢Must load data, periodically refresh it, and purge too-old data

Metadata management

➢Must keep track of source, loading time, and other information for all data in the warehouse

Data Warehouse Types

Virtual Data Warehouses

- Provide views of operational DBs that are materialized for efficiency

Data Marts

- Targeted to a subset of the organization
- Also called department-level data warehouse
- Low-risk, low-cost, but hard to evolve

Enterprise-wide Data Warehouses

- Large projects with massive investment of time and resources

Consider a table of transactions:

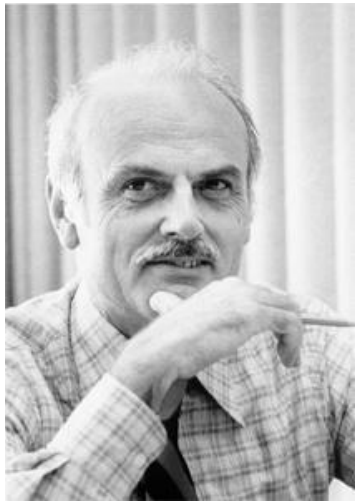
Day	Product	Store	Sales (AUD)
…	…	…	…
9/2/2014	Milk	Toowong	3412
10/2/2014	Milk	Toowong	2918
9/2/2014	Bread	Toowong	2918
10/2/2014	Bread	Toowong	3445
9/2/2014	Milk	Kenmore	5440
10/2/2014	Milk	Kenmore	4992
9/2/2014	Bread	Kenmore	2918
10/2/2014	Bread	Kenmore	3067
…	…	…	…

- Can these **facts be automatically summarized** (aggregated) in order to answer analytical queries?
  - ✓ How many different locations of Stores?
  - ✓ What kinds of Products sold well?
  - ✓ Can we get the monthly report on sales?

Multidimensional Data Model

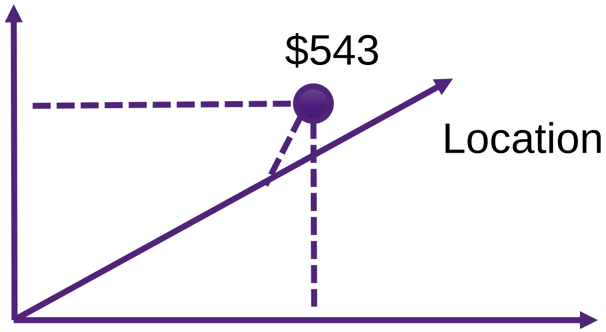
"There are typically a **number of dimensions** from which a given pool of data can be analyzed. This plural perspective, or **multidimensional conceptual view**, appears to be the way most business persons naturally view their enterprise."

- Codd 1993



Edgar Frank Codd
19/08/1923 - 18/04/2003 (aged 79)

Codd, Edgar Frank (June 1970). "A Relational Model of Data for Large Shared Data Banks". Communications of the ACM. 13 (6): 377–387.



The Fact Table

The core of a data warehouse is a fact table

The facts are the values for the object of interest

- A fact about that data entity
- Raw data to be aggregated
- There are lots of instances of these facts

Associated with each fact is a key that is used for identifying, for example, which day, which product and which store.

A **fact** can be defined by a proposition which can be read as a complete sentence.

…facts vs dimensions

Dimensions

Each key is a dimension – the example has three

Dimensions can have hierarchical organization

- ➢Days grouped into weeks, months, quarters, years
- ➢Product groups **aggregated hierarchically**
  - ✓Milk ➔ dairy ➔ perishable ➔ food
  - ✓Bread ➔ baked goods ➔ perishable ➔ food
- ➢Stores grouped into regions hierarchically
  - ✓Toowong ➔ West Brisbane ➔Brisbane ➔ QLD ➔ Australia ➔ Oceania

Dimensions organized by dimension tables

Dimension Tables

Each dimension is a projection of the fact table onto one of its keys

Day
9/2/2012
10/2/2012
…

Product
Milk
Bread
…

Store
Toowong
Kenmore
…

A data warehouse is a set of facts perceived by a number of dimensions.

Design for General Dimension Tables

Time-Period

Day	Month	Qtr	Year
9/2/2012	Feb	1	2012
10/2/2012	Feb	1	2012

Region

Store	District	Region
Toowong	West	Brisbane
Kenmore	West	Brisbane

Product

Product	Kind	Type	Class
Milk	Dairy	Perishable	Food
Bread	Bakery	Perishable	Food

The Star Schema

Time-Period

Day	Month	Qtr	Year
9/2/2012	Feb	1	2012
10/2/2012	Feb	1	2012

Region

Store	District	Region
Toowong	North	Brisbane
Kenmore	West	Brisbane

Sales

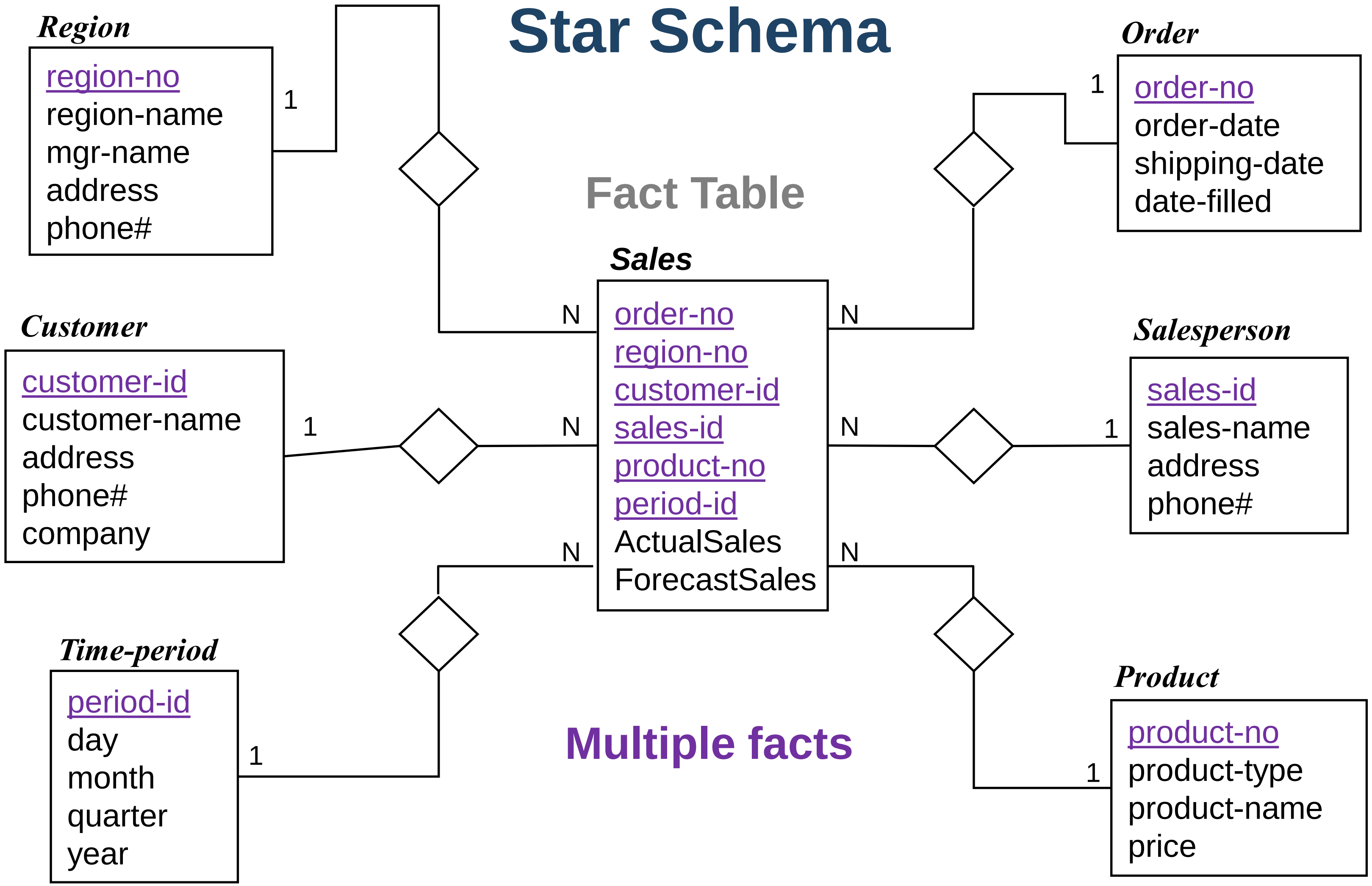
Facts

Product

Product	Kind	Type	Class
Milk	Dairy	Perishable	Food
Bread	Bakery	Perishable	Food

A fact table is much larger than dimension tables

Logical Schema (Entity Relationship)



Containment in Star Schemas

Much information stored in a containment situation

- February is in first quarter
- First quarter is in 2012, 2013…
- Dairy products are perishable
- Baked goods are perishable
- Perishable goods are food
- West is in Brisbane…

Day	Month	Qtr	Year
9/2/2012	Feb	1	2012
10/2/2012	Feb	1	2012

Facts

Day	Month
9/2/2012	Feb
10/2/2012	Feb
…	…

Month	Qtr
Feb	1
Mar	1
…	…

Qtr	Year
1	2012
2	2012
1	2013

Normalization

Many identifiers are weak

- There is a February in every year
- There is a first quarter in every year
- West in Brisbane must be distinguished from west in Sydney…

Replace weak identifiers by global identifiers in scope

- Month ID, so that Feb 2012 is M002, Feb 2013 is M014, etc
- Quarter ID, so that Q1 2012 is Q001, Q1 2013 is Q005, etc
- Brisbane West is District D13, Brisbane South D22, Sydney North is D45, etc

…weak ID: must be used with another attribute (e.g. a foreign key) in order to be able to uniquely identify an entity

Normalized Dimension Tables

Day	Month ID
9/2/2012	M002
10/2/2012	M002
…	…

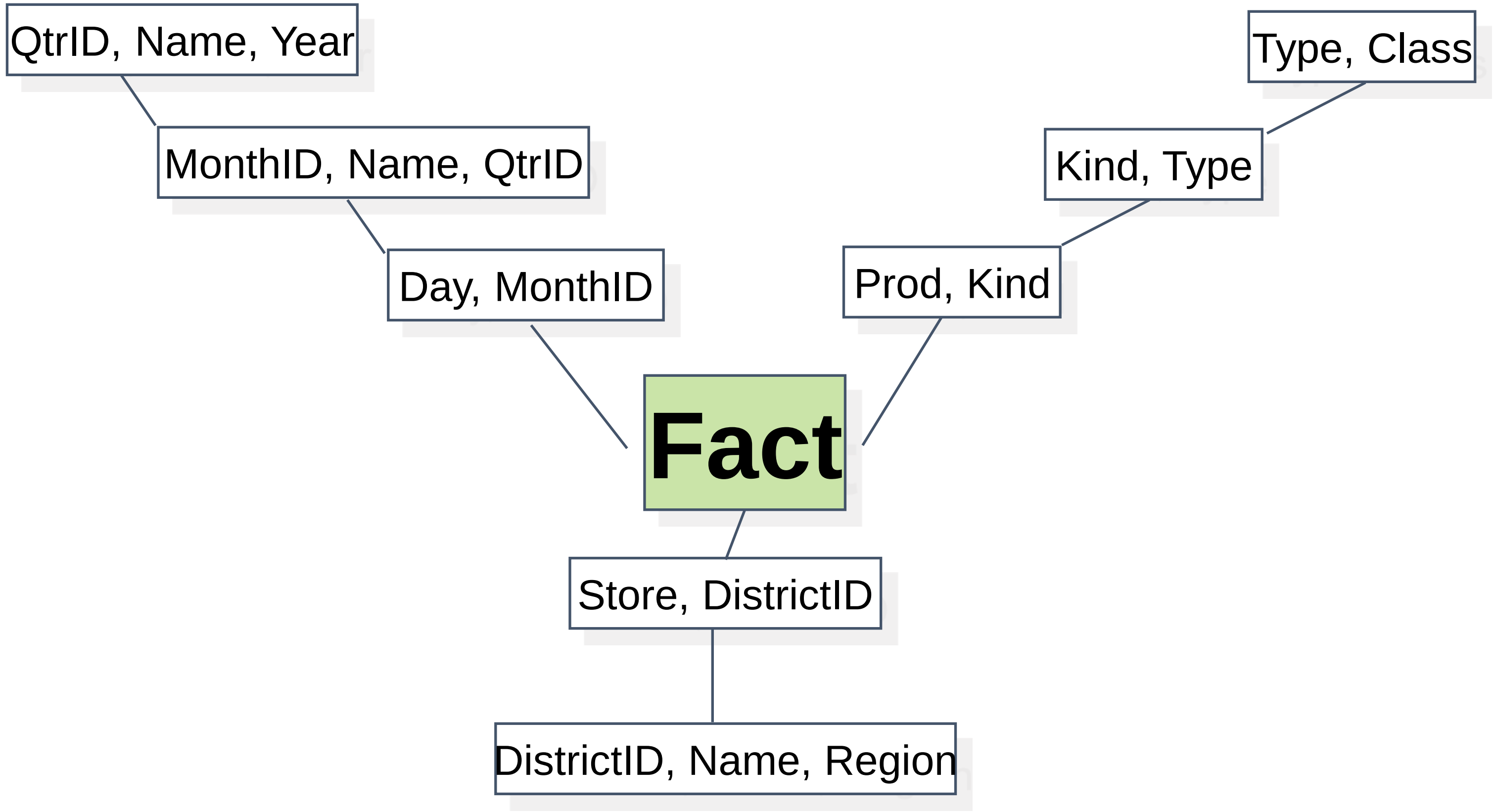
MonthID	Name	Quarter ID
M002	February	Q001
M014	February	Q005
M026	February	Q009
…	…	…

Quarter ID	Name	Year
Q001	1	2012
Q005	1	2013
Q009	1	2014
…	…	…

Original Table:

Day	Month	Qtr	Year
9/2/2012	Feb	1	2012
10/2/2012	Feb	1	2012
…	…	…	…

The Snowflake Schema



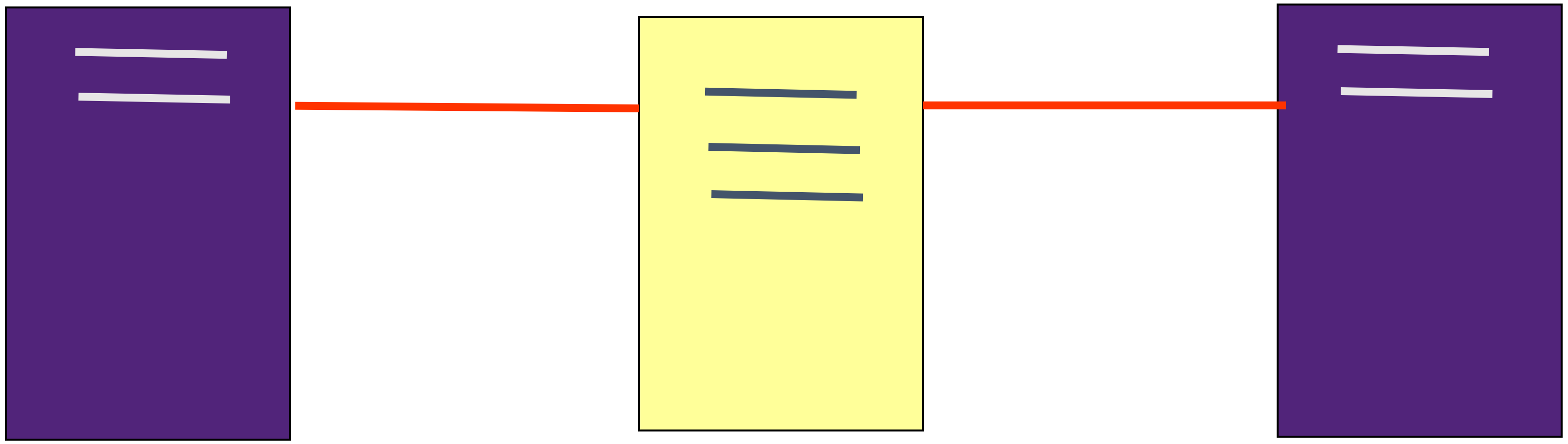
Fact Constellation

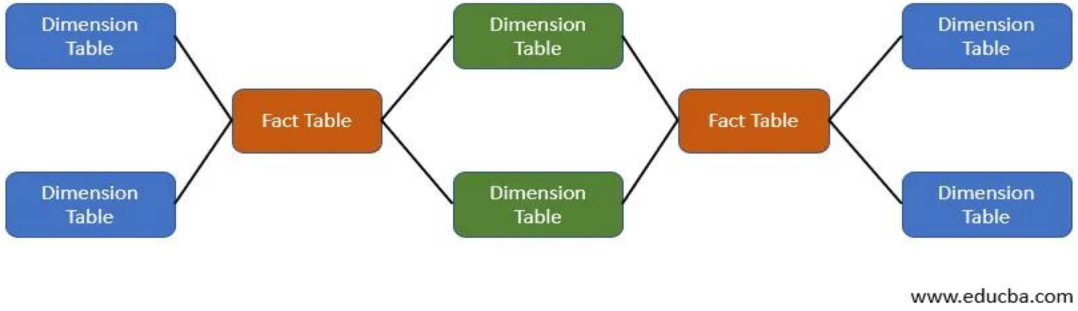
A set of fact tables that share some dimension tables

Fact table 1
Business results

Dimension Table
Product

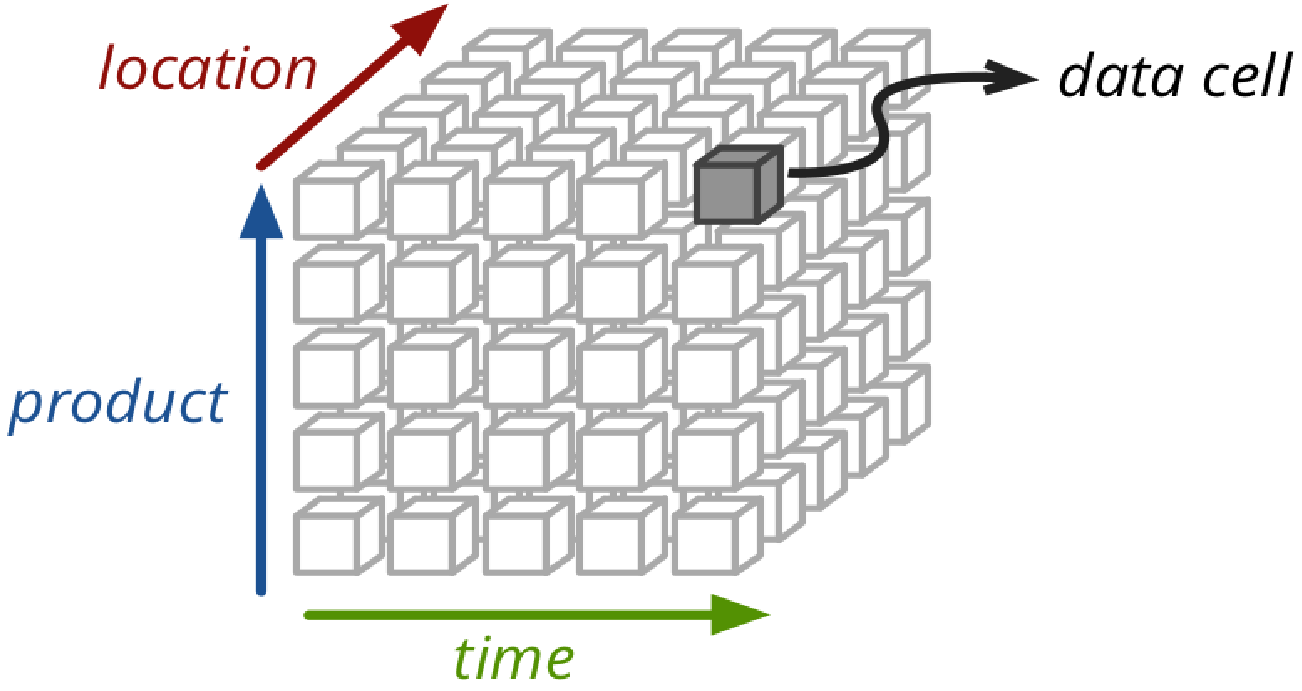
Fact table 2
Business forecast





Data Cube

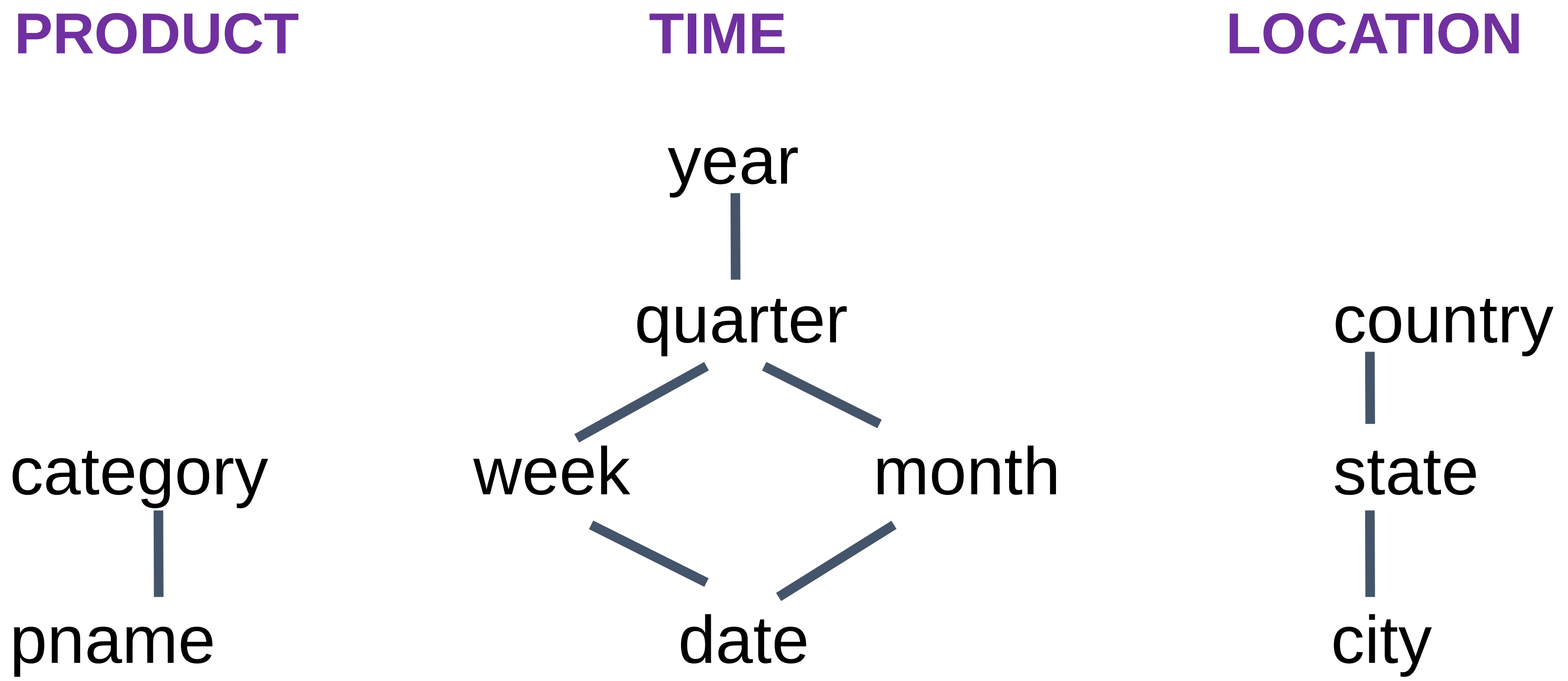
Sales data with three dimensions: location, product and time



Hypercubes if there are more than 3 dimensions

Dimension Hierarchies

Dimension Hierarchies can be defined by using **Linear**, **Tree**, or **Lattice** structures



Decision Support Systems (DSS)

Data warehousing is for Decision Support Systems (DSS)

- DSS provides decision makers in organizations with information (*data-driven decisions*)
- Queries are less well structured (for under-specified problems faced by most senior managers)
- Used by non-IT professionals (i.e., managers) interactively (*data exploration*)
- Flexible enough to accommodate changes in the environment and decision-making approaches

OLAP Queries for Decision Support

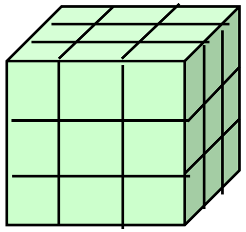
Most OLAP queries can be expressed in SQL – *this is difficult for general end users*

The goal is to give non SQL experts some tools for selected class of queries

Examples;

- find the total sales,
- find the top five products ranked by total sales,
- find total sales by month for each city,
- find % change in the total monthly sales
- for each product…

Typical Functionality of DW



Pivoting (cross-tabulation)

➢Rotate data cube to show a different orientation of axes

Roll-up

➢Move up concept hierarchy, grouping into larger units along a dimension with more generalization

Drill-down

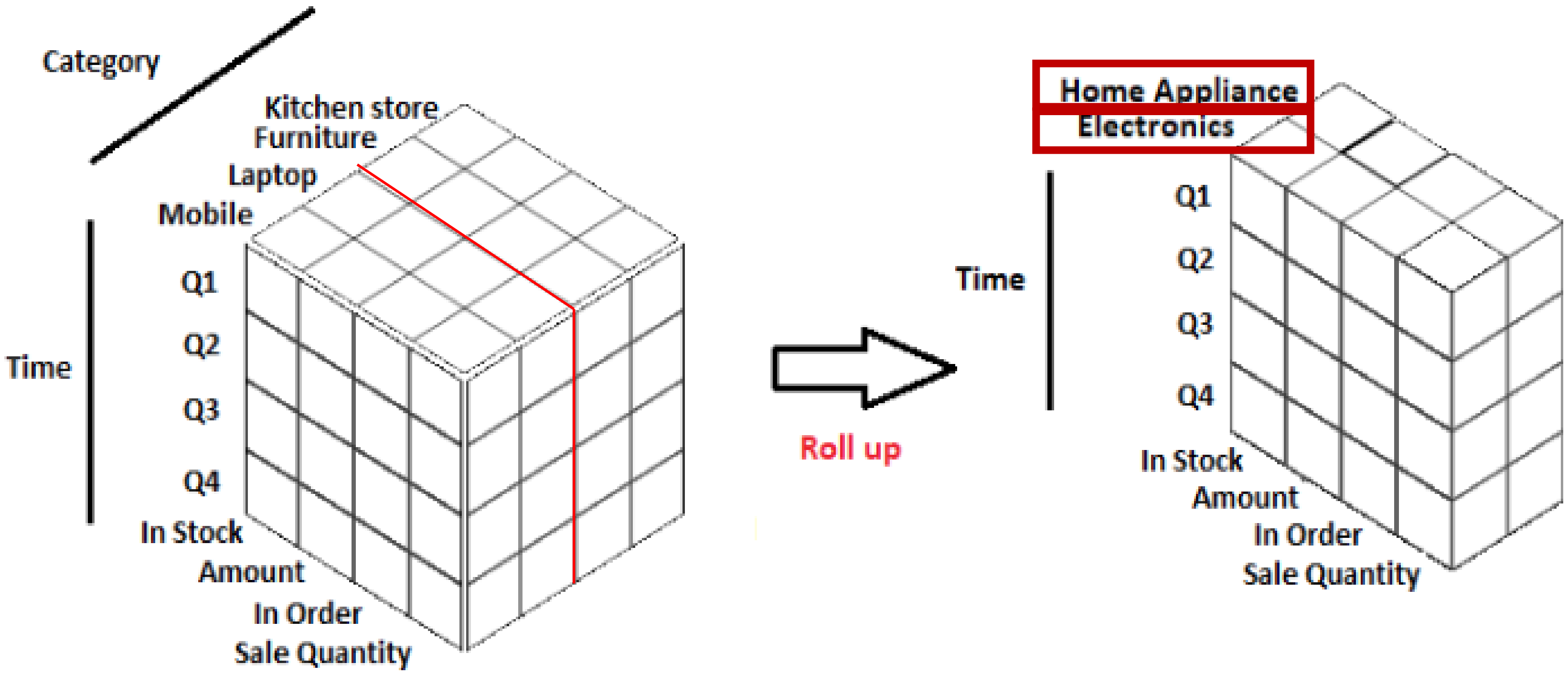
➢Disaggregate to a finer-grained view to show more details

Slice and dice

➢Perform projection operations on the dimensions

Other operations, such as arithmetic (to get derived values), sorting, selection…

An Example of Rollup



Page 147, Figure 4.12 Examples of typical OLAP operations on multidimensional data., Jiawei Han Book

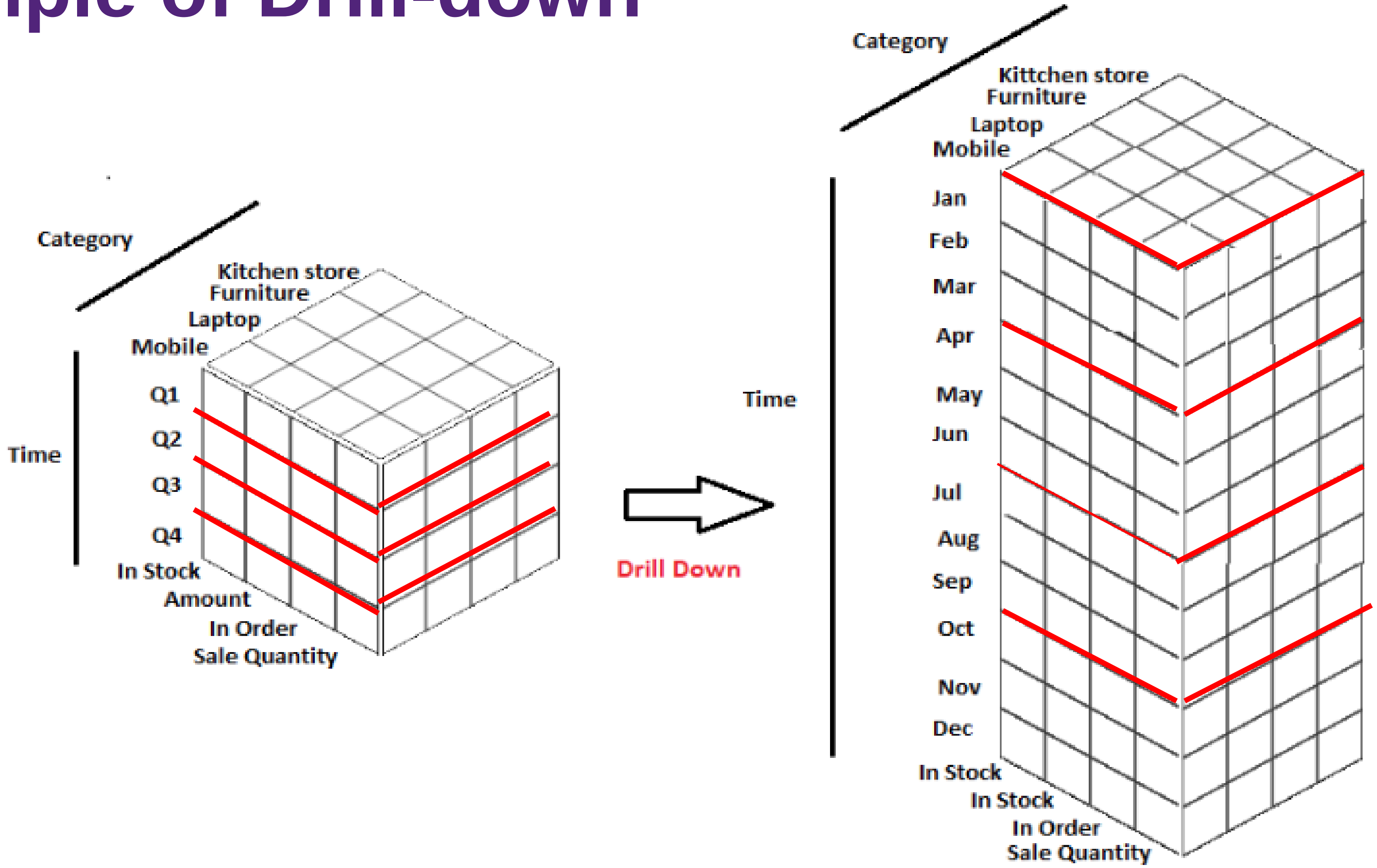
Example Roll-up

Roll-up milk, bread to compare perishables with other product groups

	Product		Total
Day	**Milk**	**Bread**	**Perishables**
9/2/2012	8952	5836	14788
10/2/2012	7910	8059	15969

	Product Group		Total
Day	**Perishables**	**Canned Goods**	**All Groups**
9/2/2012	14788	55621	206771
10/2/2012	15969	68123	310885

An Example of Drill-down



Page 147, Figure 4.12 Examples of typical OLAP operations on multidimensional data, Jiawei Han Book

Example Drill-down

Drill-down perishables to constituent products

	Product Group		Total
Day	**Perishables**	**Canned Goods**	**All Groups**
9/2/2012	14788	55621	206771
10/2/2012	15969	68123	310885
	Product		Total
Day	**Milk**	**Bread**	**Perishables**
9/2/2012	8952	5836	14788
10/2/2012	7910	8059	15969

OLAP Queries

Influenced by SQL + spreadsheets

A common operation is to aggregate a measure over one or more dimensions

- Find total sales
- Find total sales for each city, or for each state
- Find top five products ranked by total sales

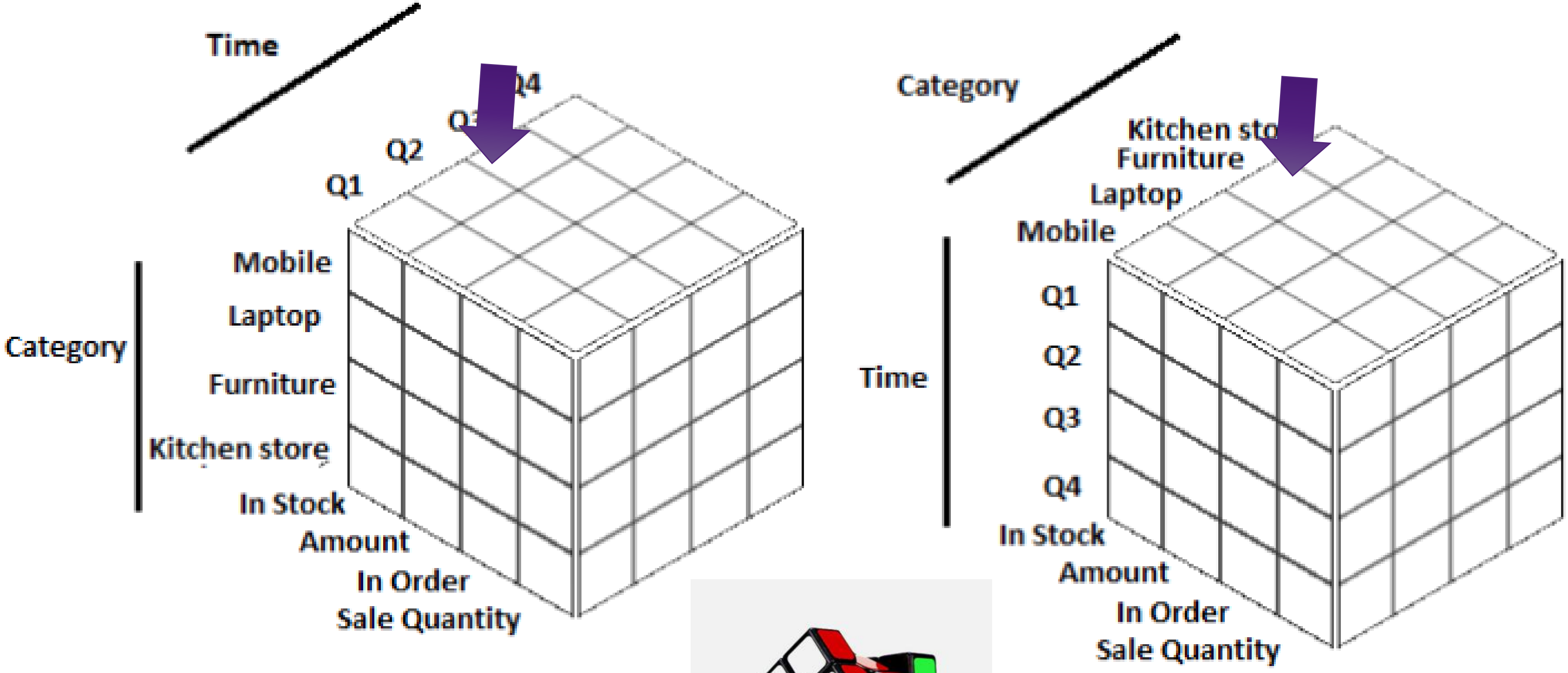
Roll-up: Aggregating at different levels of a dimension hierarchy

- Given total sales by *city*, we can roll-up to get sales by *state*

Drill-down: The inverse of roll-up

- Given total sales by *state*, can drill-down to get total sales by *city*
- Can also drill-down on different dimension to get total sales by *product* for *each state*

An Example of Pivoting



Page 147, Figure 4.12 Examples of typical OLAP operations on multidimensional data., Jiawei Han Book

Example of Pivot Query

```sql
SELECT [Year], Pankaj,Rahul,Sandeep FROM
(SELECT Name, [Year] , Sales FROM Employee )Tab1
PIVOT
(
SUM(Sales) FOR Name IN (Pankaj,Rahul,Sandeep)) AS Tab2
ORDER BY [Tab2].[Year]
```

Employee

Results | Messages

	Name	Year	Sales
1	Pankaj	2010	72500
2	Rahul	2010	60500
3	Sandeep	2010	52000
4	Pankaj	2011	45000
5	Sandeep	2011	82500
6	Rahul	2011	35600
7	Pankaj	2012	32500
8	Pankaj	2010	20500
9	Rahul	2011	200500
10	Sandeep	2010	32000

Find out the yearly sales for each Employee.

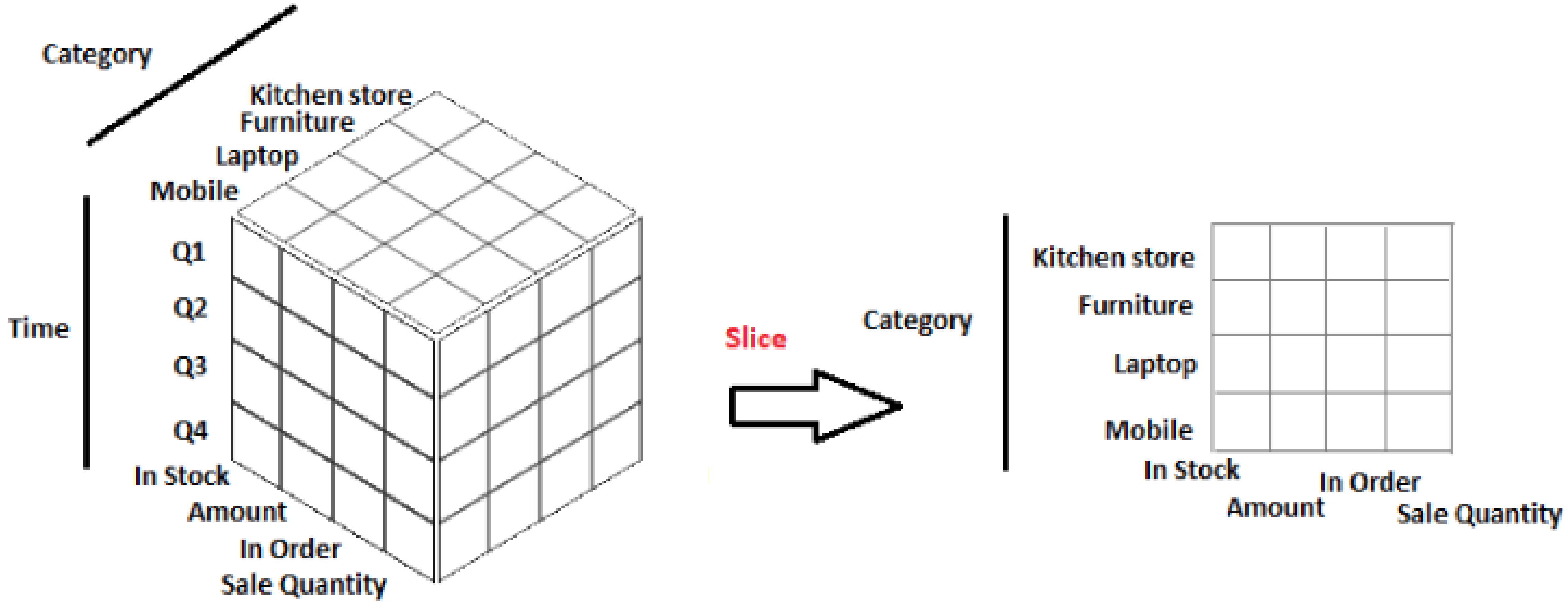
Pivot Query Output

Results | Messages

	Year	Pankaj	Rahul	Sandeep
1	2010	93000	60500	84000
2	2011	45000	236100	82500
3	2012	32500	NULL	NULL

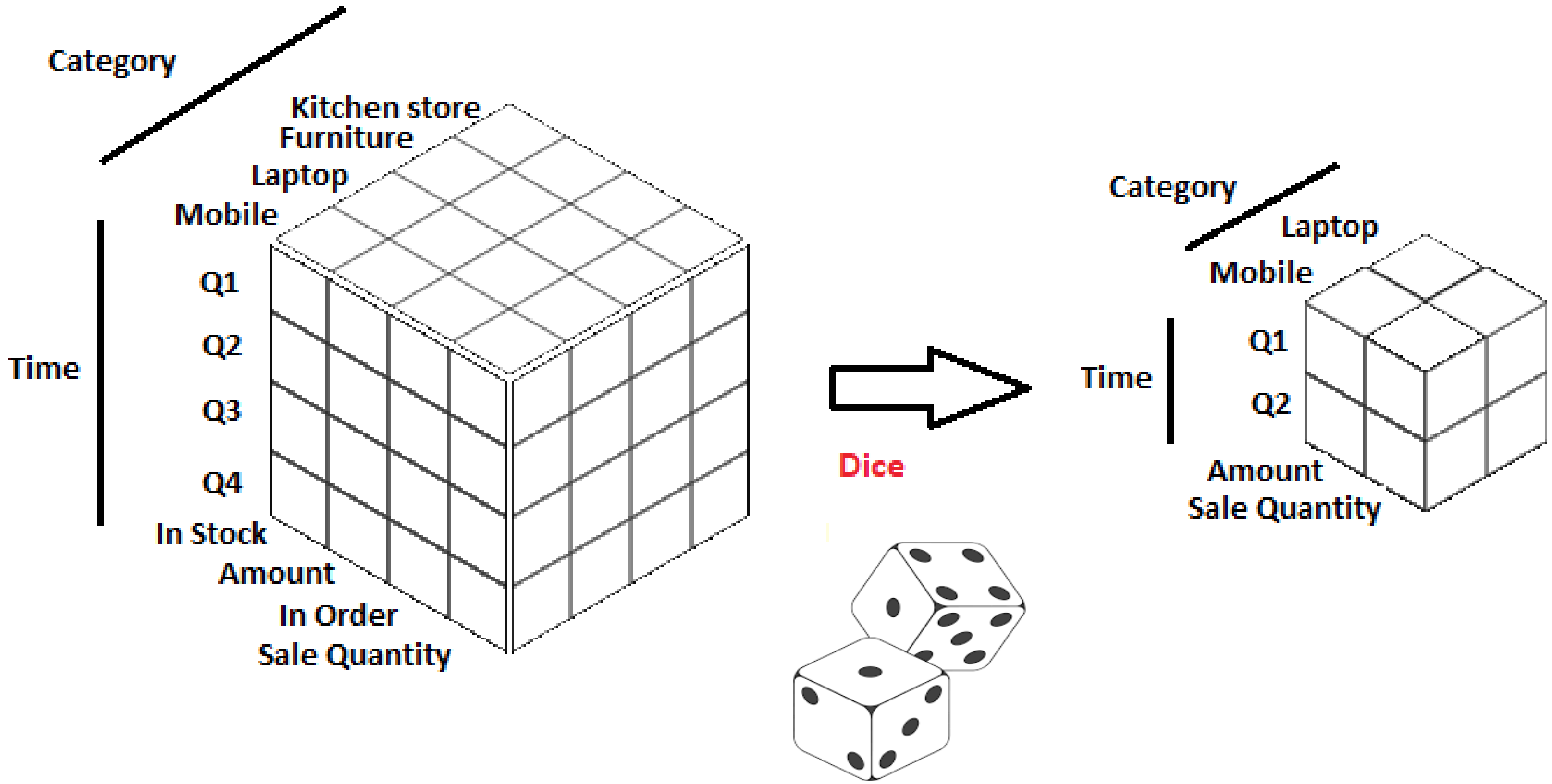
https://www.c-sharpcorner.com/UploadFile/f0b2ed/pivot-and-unpovit-in-sql-server/#:~:text=PIVOT%20relational%20operator%20converts%20data,operation%20where%20we%20need%20them.

An Example of Slicing



Page 147, Figure 4.12 Examples of typical OLAP operations on multidimensional data., Jiawei Han Book

An Example of Dicing



OLAP Queries

Slicing and Dicing: Equality and range selections on one (slice), or more (dice) dimensions

- Total of selected data behind pivot
- Similar to HAVING clause in SQL

Find the sales of products in terms of years and locations.

Pivoting: Aggregation on selected dimensions.

- E.g., Pivoting on Location and Time yields this cross-tabulation: Cells contain sums of data from other dimensions (data behind pivot)
- Metaphor of *rotating* data cube

Q2

Q1	WI	CA	Total
1995	63	81	144
1996	38	107	145
1997	75	35	110
Q3 Total	176	223	339

Pivoting by Multiple SQL Queries

The cross-tabulation obtained by pivoting can also be computed using a collection of SQL queries:

Q1:

```
SELECT SUM(S.sales)
FROM     Sales S, Times T, Location L
WHERE  S.timeid=T.timeid AND S.locid=L.locid
GROUP BY T.year, L.state
```

Find the sales of products in terms of years and locations.

Q2:

```
SELECT SUM(S.sales)
FROM    Sales S, Times T
WHERE  S.timeid=T.timeid
GROUP BY T.year
```

Q3:

```
SELECT SUM(S.sales)
FROM    Sales S, Location L
WHERE  S.locid=L.locid
GROUP BY L.state
```

Sales (S)

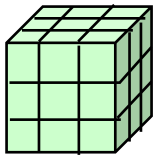
<u>timeid</u>	<u>locid</u>	<u>prodid</u>	sales

Location (L)

<u>locid</u>	country	state	city

Times (T)

<u>timeid</u>	month	Qtr	year



The CUBE Operator

Generalizing the previous example, if there are *k* dimensions, we have 2^k possible SQL **GROUP BY** queries that can be generated through pivoting on a subset of dimensions.

CUBE timeid, pid, locid BY SUM Sales

➢ Equivalent to rolling up Sales on all eight subsets of the set {**timeid, pid, locid**}

```
SELECT SUM(S.sales)
FROM    Sales S, …
GROUP BY grouping-list
```

Sales (S)

Day	Product	Store	Sales ($)

The CUBE operator has been implemented in most data warehousing products, and often used together with SQL statements following GROUP_BY. It basically creates a cube using the listed dimensions for the required aggregations (in the SUM part). For example, if CUBE(a, b, c) is used, where a, b and c are dimensions (attributes with their hierarchies), it will **generate eight (8) aggregates for all the following combinations: (a, b, c), (a, b), (a, c), (b, c), (a), (b), (c) and (null)**. That is, for k dimensions in the CUBE list, 2^k types of group-bys will be generated. Therefore, once CUBE(a, b, c) is used, aggregates based on all these 2^k dimension combinations are generated.

Review

We need Data Warehouses to provide high performance to read-only queries in Decision Support Systems, without compromising transactional operations

Building a Data Warehouse is a multi-step process that requires both organizational as well as technical support especially in ETL

Multi-dimensional Data Models have been proposed for Data Warehouses by which we can design schemas (e.g. star schema, snowflake schema) suitable for Data Warehouse operations

Data Warehouses are supported by specialized OLAP (Online Analytical Processing) Queries