EIE 3112
Data Warehousing

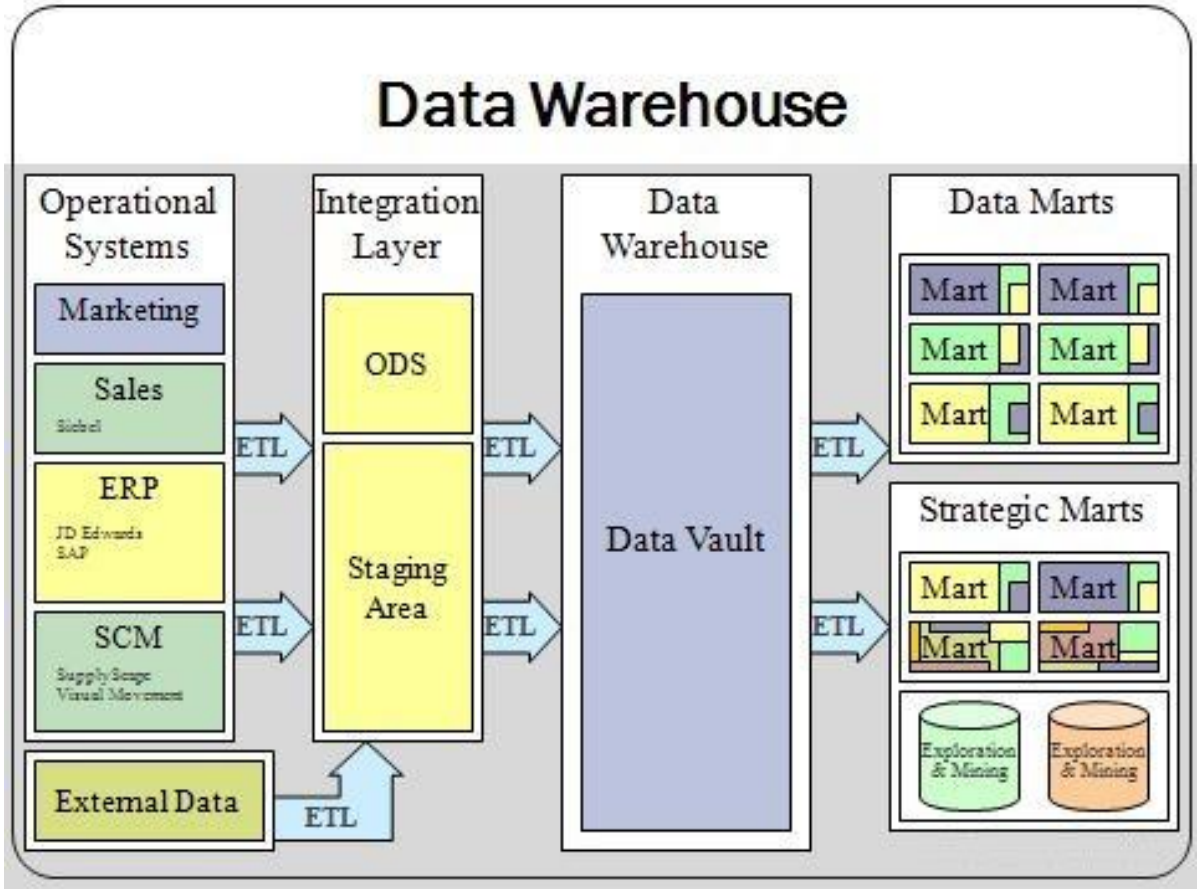
- T. Connolly and C. Begg, "*Database Systems: A Practical Approach to Design, Implementation, and Management*," 6th Edition, Chapter 31, Pearson, 2015. (5th Edition is also fine)
- http://www.codeproject.com/Articles/652108/Create-First-Data-WareHouse
- http://www.tutorialspoint.com/dwh/

Contents

- How data warehousing evolved.
- The main concepts and benefits associated with data warehousing.
- How online transaction processing (OLTP) systems differ from a data warehouse.
- The architecture and main components of a data warehouse.
- The concept of a data mart and the main reasons for implementing a data mart
- Designing data warehouses

Data Warehouse Overview

- https://www.youtube.com/watch?v=zTs5zjSXnvs



Source:
https://en.wikipedia.org/wiki/Data_warehouse

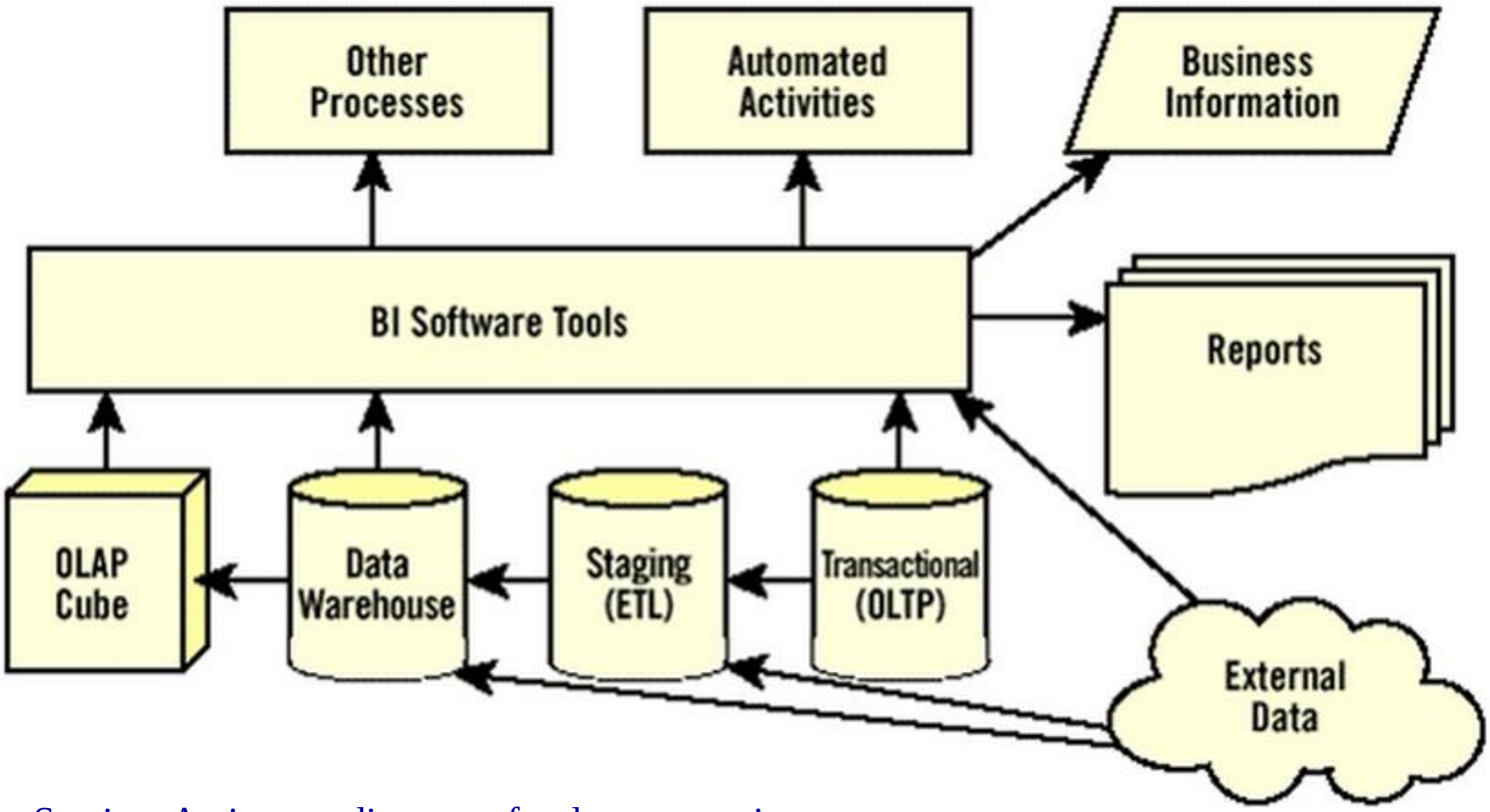
The Evolution of Data Warehousing

- Since 1970s, organizations gained competitive advantage through systems that automate business processes to offer more efficient and cost-effective services to the customer.
- This resulted in accumulation of growing amounts of data in operational databases.
- Organizations now focus on ways to use operational data to support decision-making
- This leads to the emergence of an area called **business intelligence (BI)**

The Evolution of Data Warehousing

- BI refers to the processes for collecting and analysing data with the purpose of facilitating corporate decision making.
  - A bank might analyze ATM transactions for behavior, time of day, or queue information
  - A retailer might perform analysis on its point of sale transactions.
- Data warehousing is one of the key technologies for implementing BI.

Relationship between BI and DW



Staging: An intermediate area for data processing
OLTP: Online transaction processing
OLAP: Online analytical processing

Goal of Data Warehousing

- To integrate corporate data into a single repository from which users can easily run queries, produce reports, and perform analysis (for making business decisions).

Data Warehousing Concepts

- A subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision-making process (Inmon, 1993).

Data Warehousing Concepts

- **Subject-Oriented**:
  - The warehouse is organized around the major subjects of the enterprise (e.g. customers, products, and sales) rather than the business operations (e.g. customer invoicing, stock control, and product sales).
  - This is reflected in the need to store decision-support data rather than application-oriented data.

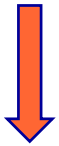
Data Warehousing Concepts

- **Integrated Data**:
  - The data warehouse integrates corporate application-oriented data from different source systems (databases), which often includes data that is inconsistent (different format).
  - The integrated data source must be made consistent to present a unified view of the data to the users (see example next page).

Data Warehousing Concepts

Example of source data:

System Name	Attribute Name	Column Name	Datatype	Values
Source System 1	Customer Application Date	CUSTOMER_APPLICATION_DATE	NUMERIC(8,0)	11012005
Source System 2	Customer Application Date	CUST_APPLICATION_DATE	DATE	11012005
Source System 3	Application Date	APPLICATION_DATE	DATE	01NOV2005



Avoid data inconsistency by integrating data from different sources into a data warehouse with consistent data type

Example of target data in data warehouse:

Target System	Attribute Name	Column Name	Datatype	Values
Record #1	Customer Application Date	CUSTOMER_APPLICATION_DATE	DATE	01112005
Record #2	Customer Application Date	CUSTOMER_APPLICATION_DATE	DATE	01112005
Record #3	Customer Application Date	CUSTOMER_APPLICATION_DATE	DATE	01112005

Data Warehousing Concepts

- **Time Variant:**
  - The data in a data warehouse is identified with a particular time period. The data in data warehouse provide information from a historical point of view.
- **Non-volatile Data**:
  - Data in the warehouse is not normally updated in real-time (RT) but is refreshed from operational systems on a regular basis. (However, emerging trend is towards RT or near RT DWs)
  - New data is always added as a supplement to the database, rather than a replacement.

Benefits of Data Warehousing

- Potential high returns on investment (ROI)
  - ROI of 431% by International Data Corporation (IDC)
- Competitive advantage
  - Better business decisions
  - Reveal of untapped information
- Increased productivity of corporate decision-makers
  - By transforming data into meaningful information, DW allows corporate decision makers to perform more accurate and consistent analysis.

OLTP versus Data Warehousing

- DBMS built for online transaction processing (OLTP) are usually not suitable for data warehousing.
- OLTP is designed to maximize transaction throughput, whereas data warehousing is designed to support ad hoc queries for business decision making.

Data Warehouse Queries

- The types of queries that a data warehouse is expected to answer ranges from the relatively simple to the highly complex and is dependent on the type of end-user access tools used.

- End-user access tools include:
  - Traditional reporting and query
  - OLAP
  - Data mining

Typical Data Warehouse Queries

As a manager of a store, you may want to ask these questions:

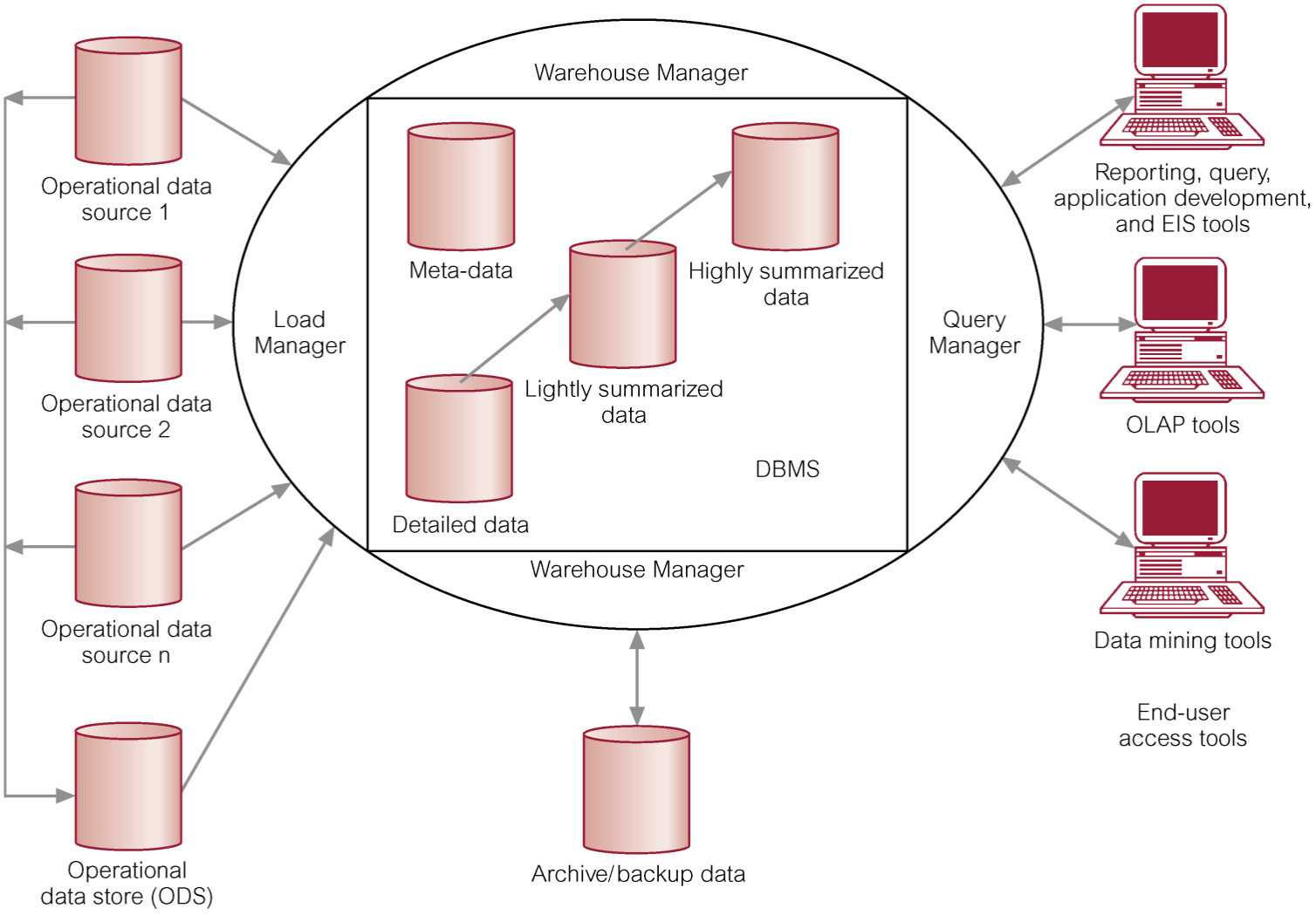
- What is the daily, weekly, monthly, quarterly profit of each store?
- Which product has more demand and in which location?
- What is the trend of sales by time period of the day over the week, month, and year?
- On what day sales is higher?
- On every Sunday of this month, what is sales and what is profit?
- What is trend of sales on weekday and weekend?

Different types of questions lead to different DW designs

Exercise

- Suppose you are senior management of the University, give an example of question that you may want to retrieve answer from Data Warehouse.

Example Data Warehouse Architecture



Operational Data Sources

- Main sources are online transaction processing (OLTP) databases.
- Also include sources such as personal databases and spread sheets, and web usage log files.
- Holds current and integrated operational data for analysis.

ETL Manager

- ETL stands for **e**xtraction, **t**ransformation, and **l**oading
- An ETL manager performs all the operations associated with the extraction and loading of data into the warehouse.
- Data for a DW must be extracted from one or more data sources, transformed into a form that is easy to analyze and consistent with data already in the warehouse, and then finally loaded into the DW.

Warehouse Manager

- Performs all the operations associated with the management of the data in the warehouse such as:
  - Analysis of data to ensure consistency.
  - Transformation and merging of source data from temporary storage into data warehouse tables.
  - Creation of indexes and views on base tables.
  - Generation of denormalizations, (if necessary).
  - Generation of aggregations, (if necessary).
  - Backing-up and archiving data.

Query Manager

- Performs the operations associated with the management of user queries such as –
  - Directing queries to the appropriate tables and scheduling the execution of queries.
  - In some cases, the query manager also generates query profiles to allow the warehouse manager to determine which indexes (for speeding up future queries) and aggregations are appropriate.

End-User Access Tools

- Main purpose of DW is to support decision makers and this is achieved through the provision of a range of access tools including:
  - reporting and querying,
  - application and development,
  - online analytical processing (OLAP),
  - data mining.

Data Warehousing Tools and technologies – ETL Processes

- *Extraction*
  - Targets one or more data sources and these sources typically include OLTP databases but can also include personal databases and spreadsheets, Enterprise Resource Planning (ERP) files, and web usage log files.
  - The data sources are normally internal but can also include external sources such as the systems used by suppliers and/or customers.

Data Warehousing Tools and technologies – ETL Processes

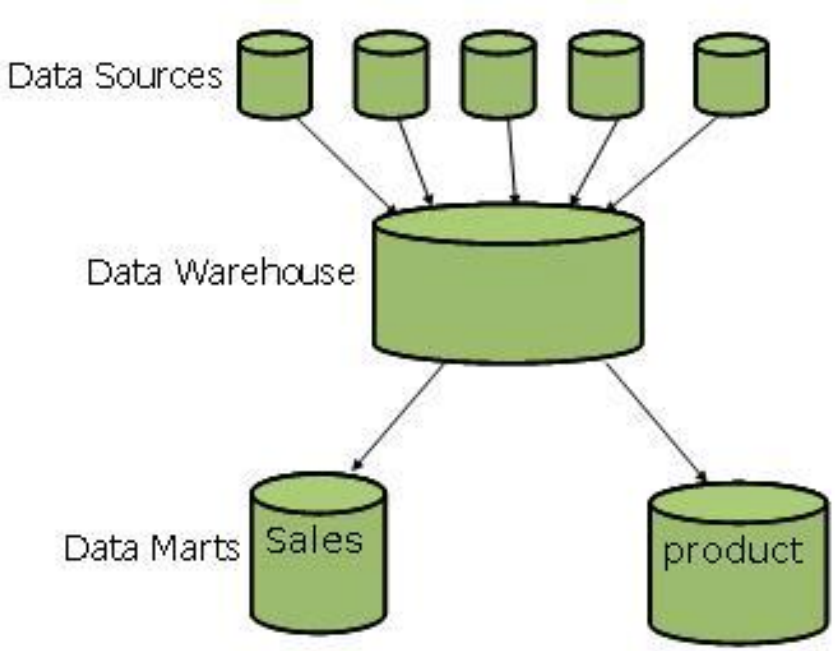
- *Transformation*
  - Applies a series of rules or functions to the extracted data, which determines how the data will be used for analysis and can involve transformations such as data summations, data encoding, data merging, data splitting, data calculations, and creation of surrogate keys.

Data Warehousing Tools and technologies – ETL Processes

- *Loading*
  - Data Loading involves sorting, summarizing, consolidating, checking integrity and building indices and partitions (for speeding up queries).

Data Mart

- A data mart is a database that contains a subset of corporate data to support the analytical requirements of a particular business unit (such as the Sales department) or to support users who share the same requirements to analyse a particular business process (such as property sales).



Reasons for Creating a Data Mart

- To give users access to the data they need to analyze most often.
- To provide data in a form that matches the collective view of the data by a group of users in a department or business application area.
- To improve end-user response time due to the reduction in the volume of data to be accessed.
- To provide appropriately structured data as dictated by the requirements of the end-user access tools.

Reasons for Creating a Data Mart

- Building a data mart is simpler compared with establishing an enterprise-wide DW (EDW).
- The cost of implementing data marts is normally less than that required to establish a EDW.
- The future users of a data mart are more easily defined and targeted to obtain support for a data mart than an enterprise-wide data warehouse project.

Data Warehousing Design

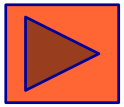
- **Identify and Collect Requirements:** Interview the key decision makers to know
  - what factors define the success in the business
  - how does management want to analyze their data
  - what are the most important business questions
- These questions will result in a number of possible queries that the DW should support, e.g., see this slide 

Typical Data Warehouse Queries

As a manager of a store, you may want to ask these questions:

- What is the daily, weekly, monthly, quarterly profit of each store?
- Which product has more demand and in which location?
- What is the trend of sales by time period of the day over the week, month, and year?
- On what day sales is higher?
- On every Sunday of this month, what is sales and what is profit?
- What is trend of sales on weekday and weekend?

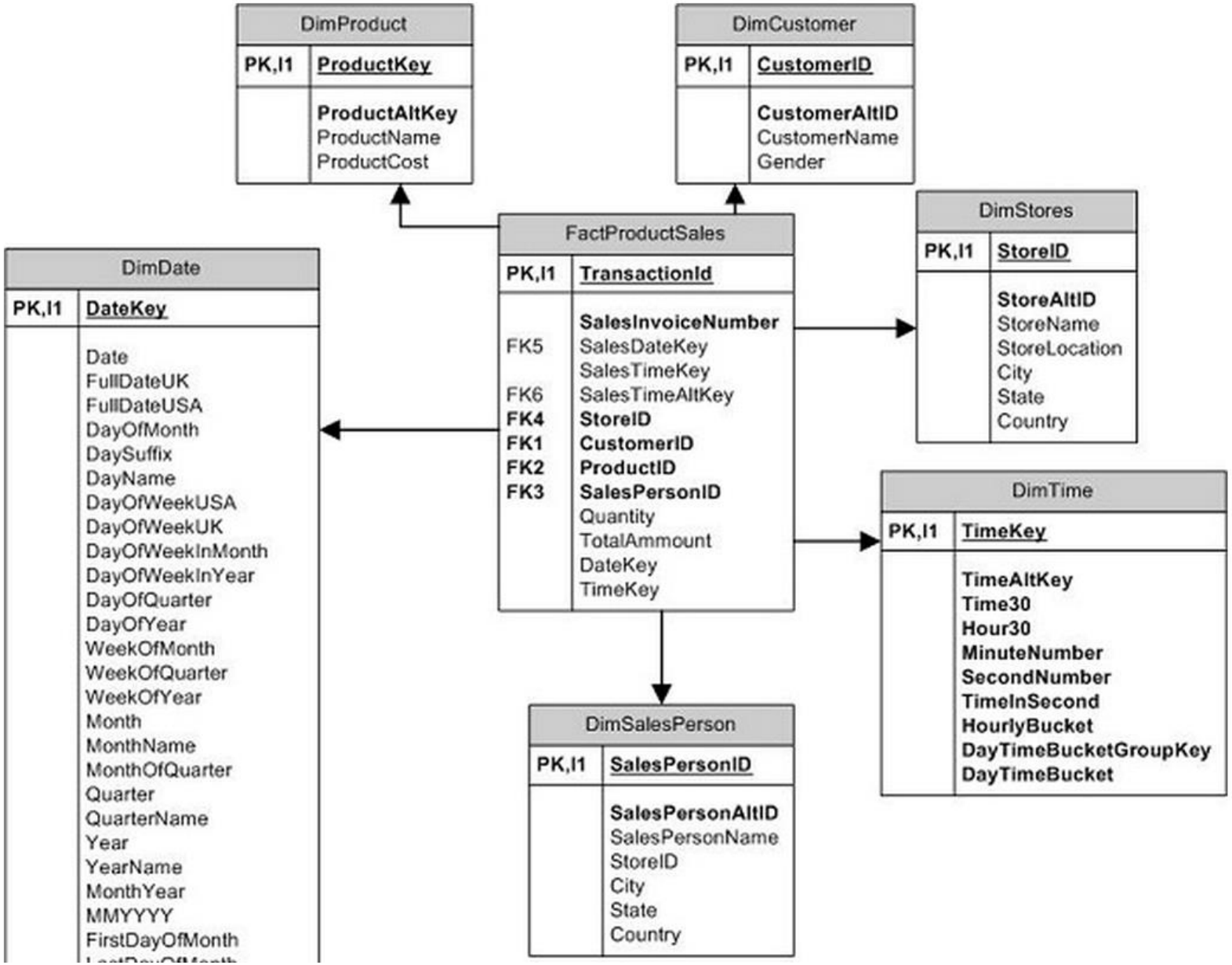
Different types of questions lead to different DW designs



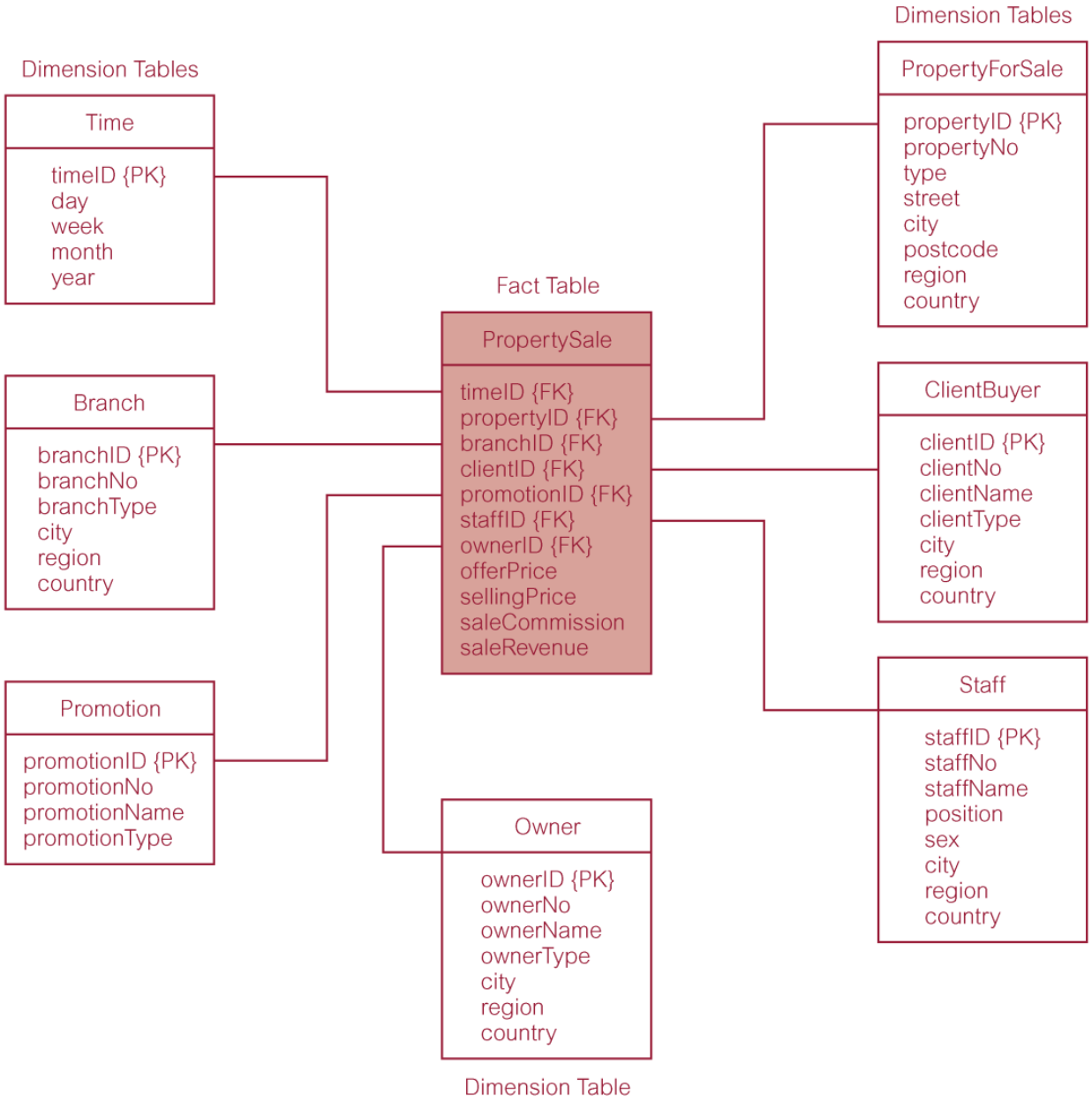
Data Warehousing Design

- **Design the Dimensional Model (star schema):**
  - A dimensional model comprises a **fact** table and a number of **dimension** tables
  - The fact table contains measures, which are historical transactional entries of the operational database, e.g., sales amount
  - Dimension tables contain textual descriptions about the subjects of the business.
  - The design is to provide instantaneous query results for analysts.

Dimensional Model (Example 1)



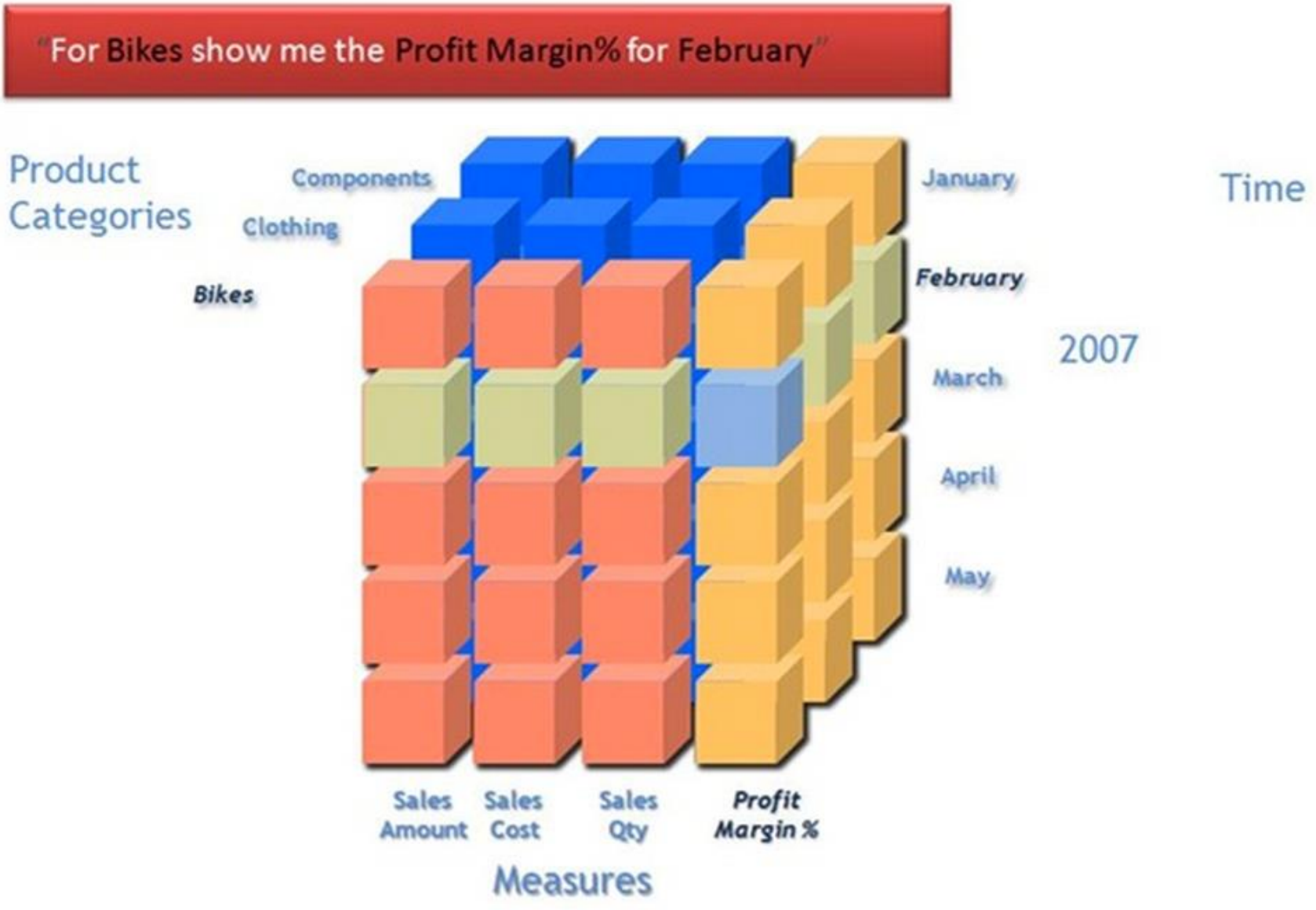
Dimensional Model (Example 2)



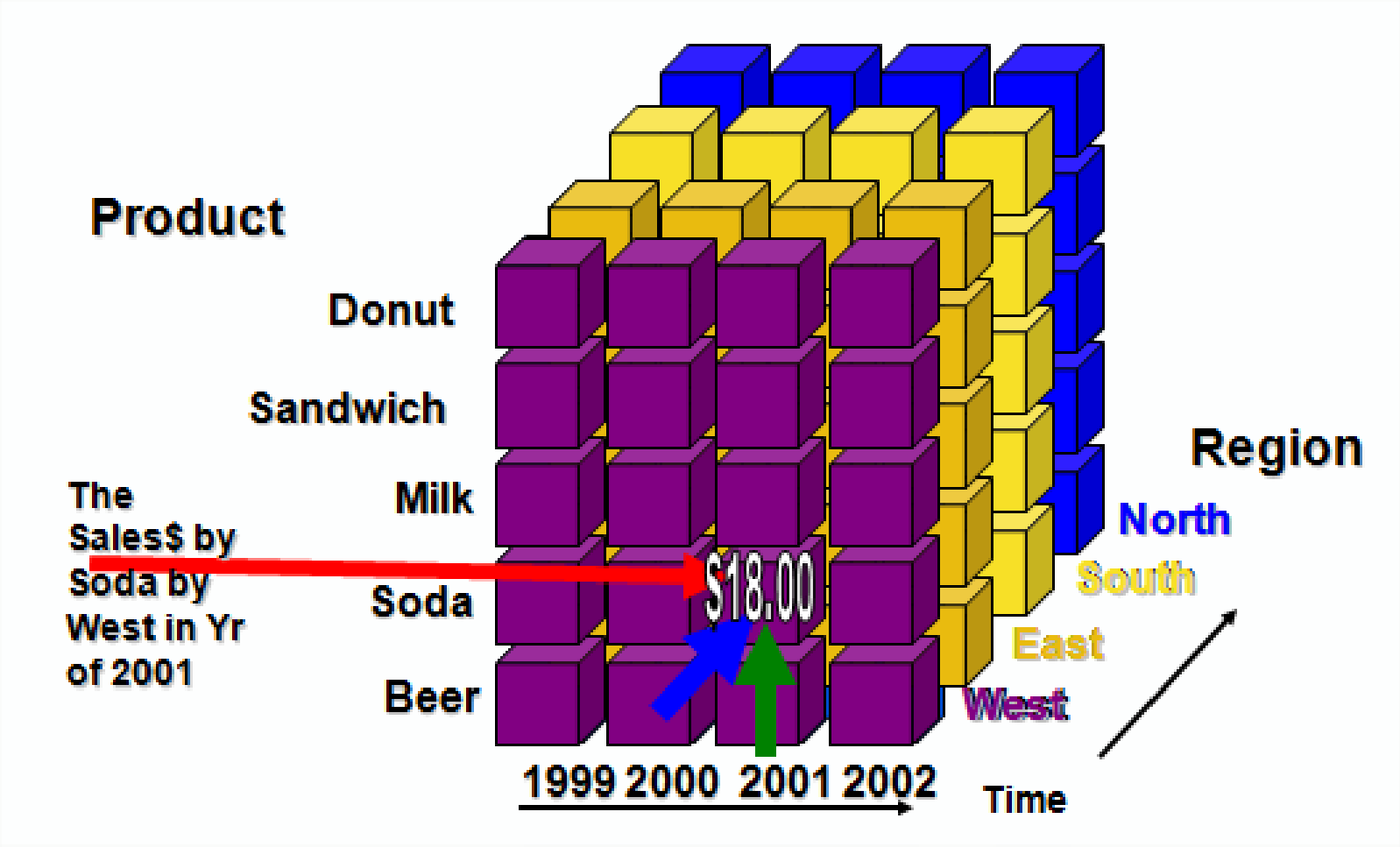
Data Warehousing Design

- **Create OLAP Cube:**
  - An OLAP cube stores data in an optimized way to provide a quick response to various types of complex queries by using dimensions and measures.
  - Most cubes store pre-aggregates of the measures with its special storage structure to provide quick response to queries.
  - Once the cube is created, we may perform queries on it instead of on the original data sources.

OLAP Cube (Example 1)



OLAP Cube (Example 2)



MySQL: With Rollup

The `GROUP BY` clause permits a `WITH ROLLUP` modifier that causes summary output to include extra rows that represent higher-level (that is, super-aggregate) summary operations. `ROLLUP` thus enables you to answer questions at multiple levels of analysis with a single query. For example, `ROLLUP` can be used to provide support for OLAP (Online Analytical Processing) operations.

Suppose that a `sales` table has `year`, `country`, `product`, and `profit` columns for recording sales profitability:

```
1  CREATE TABLE sales
2  (
3      year    INT,
4      country VARCHAR(20),
5      product VARCHAR(32),
6      profit  INT
7  );
```

https://dev.mysql.com/doc/refman/5.7/en/group-by-modifiers.html

MySQL: With Rollup

To summarize table contents per year, use a simple `GROUP BY` like this:

```
mysql> SELECT year, SUM(profit) AS profit
       FROM sales
       GROUP BY year ASC;
+------+--------+
| year | profit |
+------+--------+
| 2000 |   4525 |
| 2001 |   3010 |
+------+--------+
```

The output shows the total profit for each year. To also determine the total profit summed over all years, you must add up the individual values yourself or run an

Adding a `WITH ROLLUP` modifier to the `GROUP BY` clause causes the query to produce another row that shows the grand total over all year values:

```
mysql> SELECT year, SUM(profit) AS profit
       FROM sales
       GROUP BY year ASC WITH ROLLUP;
+------+--------+
| year | profit |
+------+--------+
| 2000 |   4525 |
| 2001 |   3010 |
| NULL |   7535 |
+------+--------+
```

The `NULL` value in the `year` column identifies the grand total super-aggregate line.

MySQL: With Rollup

`ROLLUP` has a more complex effect when there are multiple `GROUP BY` columns. In this case, each time there is a change in value in any but the last grouping column, the query produces an extra super-aggregate summary row.

For example, without `ROLLUP`, a summary of the `sales` table based on `year`, `country`, and `product` might look like this, where the output indicates summary values only at the year/country/product level of analysis:

```
mysql> SELECT year, country, product, SUM(profit) AS profit
       FROM sales
       GROUP BY year ASC, country ASC, product ASC;
+------+---------+------------+--------+
| year | country | product    | profit |
+------+---------+------------+--------+
| 2000 | Finland | Computer   |   1500 |
| 2000 | Finland | Phone      |    100 |
| 2000 | India   | Calculator |    150 |
| 2000 | India   | Computer   |   1200 |
| 2000 | USA     | Calculator |     75 |
| 2000 | USA     | Computer   |   1500 |
| 2001 | Finland | Phone      |     10 |
| 2001 | USA     | Calculator |     50 |
| 2001 | USA     | Computer   |   2700 |
| 2001 | USA     | TV         |    250 |
+------+---------+------------+--------+
```

MySQL: With Rollup

With `ROLLUP` added, the query produces several extra rows:

```
mysql> SELECT year, country, product, SUM(profit) AS profit
       FROM sales
       GROUP BY year ASC, country ASC, product ASC WITH ROLLUP;
+------+---------+------------+--------+
| year | country | product    | profit |
+------+---------+------------+--------+
| 2000 | Finland | Computer   |   1500 |
| 2000 | Finland | Phone      |    100 |
| 2000 | Finland | NULL       |   1600 |
| 2000 | India   | Calculator |    150 |
| 2000 | India   | Computer   |   1200 |
| 2000 | India   | NULL       |   1350 |
| 2000 | USA     | Calculator |     75 |
| 2000 | USA     | Computer   |   1500 |
| 2000 | USA     | NULL       |   1575 |
| 2000 | NULL    | NULL       |   4525 |
| 2001 | Finland | Phone      |     10 |
| 2001 | Finland | NULL       |     10 |
| 2001 | USA     | Calculator |     50 |
| 2001 | USA     | Computer   |   2700 |
| 2001 | USA     | TV         |    250 |
| 2001 | USA     | NULL       |   3000 |
| 2001 | NULL    | NULL       |   3010 |
| NULL | NULL    | NULL       |   7535 |
+------+---------+------------+--------+
```

Now the output includes summary information at four levels of analysis, not just one:

Exercise:
Draw the OLAP Cube for this query

```
mysql> SELECT year, country, product, SUM(profit) AS profit
       FROM sales
       GROUP BY year ASC, country ASC, product ASC;
+------+---------+-----------+--------+
| year | country | product   | profit |
+------+---------+-----------+--------+
| 2000 | Finland | Computer  |   1500 |
| 2000 | Finland | Phone     |    100 |
| 2000 | India   | Calculator|    150 |
| 2000 | India   | Computer  |   1200 |
| 2000 | USA     | Calculator|     75 |
| 2000 | USA     | Computer  |   1500 |
| 2001 | Finland | Phone     |     10 |
| 2001 | USA     | Calculator|     50 |
| 2001 | USA     | Computer  |   2700 |
| 2001 | USA     | TV        |    250 |
+------+---------+-----------+--------+
```

Exercise: Draw the OLAP Cube

Please fill in the blanks in ascending order according to the arrow direction.
(Arrow head: largest value; arrow tail: smallest value)