**Data modeling**

Considering business goals, both reports can form dimensional modeling which consists of fact tables and dimension tables for analysis. As the subject of our analysis is sales, the measurement in the fact table could be the number sold. It can be analyzed by dimension columns including temperature, item name, average change in sales when colder, average change in sales when warmer.

For example, the total and average sale can be calculated per item, temperature level, temperature changes level and so on. They are intuitively understandable.

**Data partitioning**

Our weather and sales data can be partitioned by date. It can benefit analytical questions involving time range and data backfill problems. It is the nature of ETL work since new data partition will be prepared for each day.

**Data backfill**

To process the past data, it is necessary to partition data by date, so we can perform the same processes over different partitions and simplify our work in case of large historical data.

**Tradeoffs**

Normalized tables (weather, sales) will be achieved after ETL systems. At the aspect of advantages, normalized tables can bring more simple schemas, standardized data, and less redundancy, so that it will have less time for data processing.

At the aspect of disadvantages, it will cause more difficulty for dealing with data relations and require more complex query structures (lots of joins to deformalize tables). Also, it will heavy ETL pipeline maintenance.

**Assumptions**

* There are enough data request times, file or database space before running the ETL system
* Keep the following constant:
* Data request requirements
* The goal and needs of ETL system
* data description, meaning, quality

**Tools for ETL pipeline**

Hand-Coded: build ETL through scripting language like Python, it provides more flexibility and approach

Tool-Based: CloverETL, could be used to clean, transform, and migrate data; Pentaho, could be used to integrate data