Assessment 2: The Data Warehouse

CP5806 - Data and Information: Management, Security, Privacy and Ethics

Nikki Fitzherbert 13848336

Step One: Business Scenario

Northwind Equestrian is a chain of stores across New South Wales and the Australian Capital Territory selling equestrian tack and apparel to amateur equestrians and primarily caters to the English disciplines of show jumping, dressage and eventing.

Since business processes are the operational activities performed by an organisation (Kimball & Ross, 2013), for Northwind Equestrian this includes the purchase and shipment of products from suppliers, inventory management and customer sales transactions. Northwind Equestrian also has administrative functions that relate to business financial and employee management.

The company's primary financial objective is to ensure that it makes a profit on a quarterly and annual basis on both an overall and individual store level. In order to do so, it needs to be able to analyse data relating to its sales transactions and therefore this is the business process for which the company is designing and building a data warehouse.

Northwind Equestrian already collects information on daily sales transactions in each of its stores and is able to identify for each transaction, the customer, the products(s) sold, the employee that facilitated the transaction. Under each of those business dimensions, the company collects information on:

- Customer names and contact details
- Product names, brands and categories
- Store names, sizes and location types (that is, street front or shopping centre)
- Store locations down to postcode level (that is, where the sale occurred)
- Employee names and job level details and
- Transaction dates including day, month, quarter and year.

In order to be able to determine business profitability, Northwind Equestrian records the listed unit sale price of each product (the unit sales price), the amount of Goods and Services Tax (GST) charged, the discount amount (if any) received by the customer, the quantity of each product sold and the cost of obtaining each product from the supplier (the unit sales cost).

Step Two: Information Package Diagram

According to Ponniah (2010) information package diagrams are useful for determining and recording the information requirements for a data warehouse. The business scenario in the previous step identified the subject (retail sales), the metrics that Northwind Equestrian wished to analyse, the business dimensions along which these metrics would be analysed, and a summary of the attributes within each of those dimensions.

Figure 1 depicts the complete information package diagram for Northwind Equestrian's retail sales business process. It names the subject of the diagram, the metrics (facts) used to assess sales performance, the six dimensions and the exact three to five attributes under each dimension. Wherever possible, the attributes have been ordered in terms of their concept hierarchy; from most to least granular.

Information subject: Retail sales

Hierarchies/Categories	**Date**	**Product**	**Customer**	**Employee**	**Store**	**Location**
	Day	SKU	ID	ID	Name	Postcode
	Month	Name	First name	First name	Size	Suburb
	Quarter	Brand	Last Name	Last name	Location Type	City or town
	Year	Sub-category	Email	Job title		State or territory
		Category	Phone number	Job level		
	Facts: Unit sale price, GST amount, Discount amount, Unit sale cost, Quantity sold					

Figure 1: Information package diagram

Step Three: Data Design

The next step is to use the information package diagram to construct the fact and dimension tables. In general, there is a single fact table per business process, which contains the numeric measures produced by an operational measurement event; that is, the facts identified in the information package diagram. In contrast, there are many dimension tables (one for each column of the information package diagram). The dimension tables are the primary target of constraints and grouping specifications from queries and business intelligence applications (Kimball & Ross, 2013).

Each dimension table has a single primary key, which is used to uniquely identify each data record in that table. Whilst it is possible to use the natural or business key, such as customer id or a product's SKU, these are subject to business rules outside the control of the data warehouse system and the format may change or the same identifier used in multiple instances over time. Therefore, a surrogate key has been created for each table.[1]

In the fact table, the primary key can either be a concatenation of all the foreign keys from the associated dimension tables or a surrogate key. The use of a surrogate key can be useful if there are many dimensions as a concatenated key could become complicated and unwieldy. Given the presence of six dimensions, and the potential later integration of this data warehouse with other Northwind Equestrian systems, the latter approach was used in the design.

Each of the dimension and fact tables are shown in figure 2 including identification of each table's primary key, foreign key(s) where applicable) and natural key (if one exists).

Fact table

Table name: Retail sales facts

Retail sales key	Primary key
Date	Foreign key
Product key	Foreign key
Store key	Foreign key
Location key	Foreign key
Customer key	Foreign key
Employee key	Foreign key
Unit sale price	Metric
GST amount	Metric
Discount amount	Metric
Quantity sold	Metric
Unit sale cost	Metric

[1] Note that according to Kimball & Ross (2013), the only dimension exempt from the need to use a surrogate key is the time dimension as it is usually highly predictable and stable.

Dimension tables

Table name: Date dimension

Date key	Primary key
Date day	Attribute
Date month	Attribute
Date quarter	Attribute
Date year	Attribute

Table name: Store dimension

Store key	Primary key
Store name	Attribute (Natural key)
Store type	Attribute
Store size	Attribute

Table name: Product dimension

Product key	Primary key
Product SKU	Attribute (Natural key)
Product name	Attribute
Product brand	Attribute
Product sub-category	Attribute
Product category	Attribute

Table name: Location dimension

Location key	Primary key
Location postcode	Attribute
Location suburb	Attribute
Location city or town	Attribute
Location state or territory	Attribute

Table name: Customer dimension

Customer key	Primary key
Customer ID	Attribute (Natural key)
Customer first name	Attribute
Customer last name	Attribute
Customer email	Attribute
Customer phone number	Attribute

Table name: Employee

Employee key	Primary key
Employee ID	Attribute (Natural key)
Employee first name	Attribute
Employee last name	Attribute
Employee job title	Attribute
Employee job level	Attribute

Figure 2: Fact and dimension tables

Step Four: Dimensional Modelling

The next step in designing the data warehouse is the construction of an appropriate dimensional model. A star schema is typically used because the connection of every dimension table with the fact table through a single one-to-many join means that it is optimised for querying and analysis. However, a snowflake schema is sometimes used when storage space is an issue because the dimension tables are normalised. As is evident from figure 3, a star schema has been used to connect the fact and dimension tables in this scenario.

The primary and foreign keys for each table are indicated by the golden key and red diamond symbols respectively.

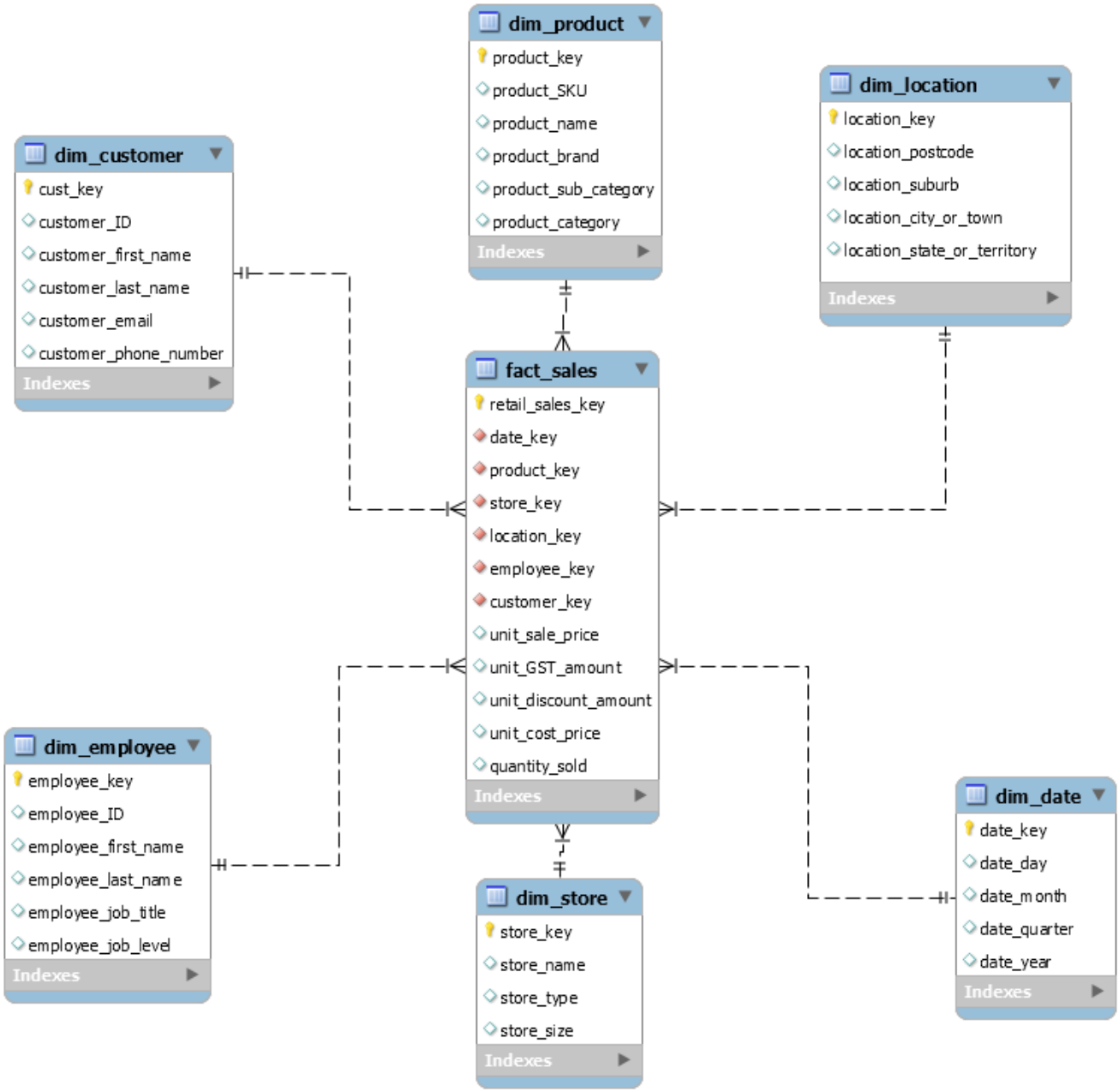


Figure 3: Star schema

Step Five: Size of the Fact Table

Part A

Fact tables store records at the most granular level and can therefore become very long as the number of time-variant records increases due to the passage of time. In fact, the number of potential rows in a fact table is equal to the product of all possible values for each dimension associated with that fact table (Ponniah, 2010).

If it assumed that Northwind Equestrian has a single year of daily sales data comprised of:

- 400 active products
- five stores in five different locations
- 55 customers and
- ten employees per store

Then the number of base fact table records could grow to:
$(1 \times 365) \times 400 \times 5 \times 5 \times 55 \times (10 \times 5) = 10.04 \approx 10$ billion records.

Part B

Such a large table can reduce query performance, particularly if those queries are complex. Pre-calculated aggregate tables can improve performance as they summarise the fact table data at higher levels along dimension table categories or hierarchies and therefore contain far fewer rows of data.

The following example illustrates how the same fact table from part A can be aggregated and reduced to approximately one million records.

Suppose that Northwind Equestrian wishes to create a fact table by aggregating along higher levels of the product, store, location and employee dimensions. Figure 4 indicates the dimension, aggregating attribute, and number of distinct values in that attribute.

Dimension	Aggregating attribute	Number of distinct values
Date	-	365
Product	Category	5
Store	Type	2
Location	State or territory	2
Customer	-	55
Employee	Job level	3

Figure 4: Fact table aggregation

The number of records in the aggregate fact table therefore reduces to:
$(1 \times 365) \times 5 \times 2 \times 2 \times 55 \times 3 = 1.20 \approx 1$ million records.

Step Six: Aggregating the Fact Table

According to Ponniah (2010), multi-way aggregate tables are formed by moving up the concept hierarchy of one or more dimensions in the dimensional model. The degree of aggregation is defined by the number of dimensions that are allowed to move from their most granular level. That is, one-way aggregation means that all but one dimension is kept at its lowest level in the hierarchy; two-way aggregation allows two dimensions to rise to higher levels and so on. Figure 5 illustrates the hierarchies along the date, product, location and store dimensions from the Northwind Equestrian dimensional model and figure 6 provides examples of one- to four-way aggregate tables using those dimensions.

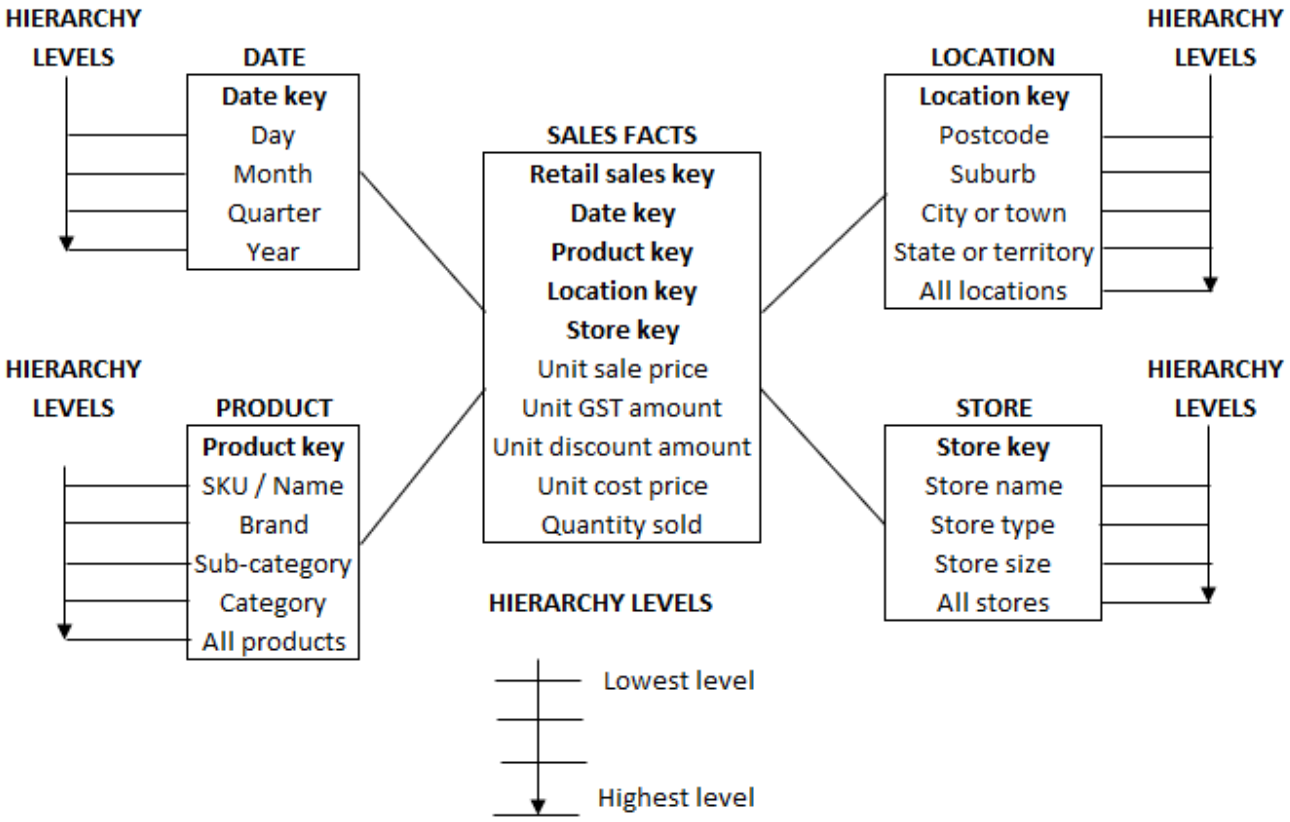


Figure 5: Dimension hierachies

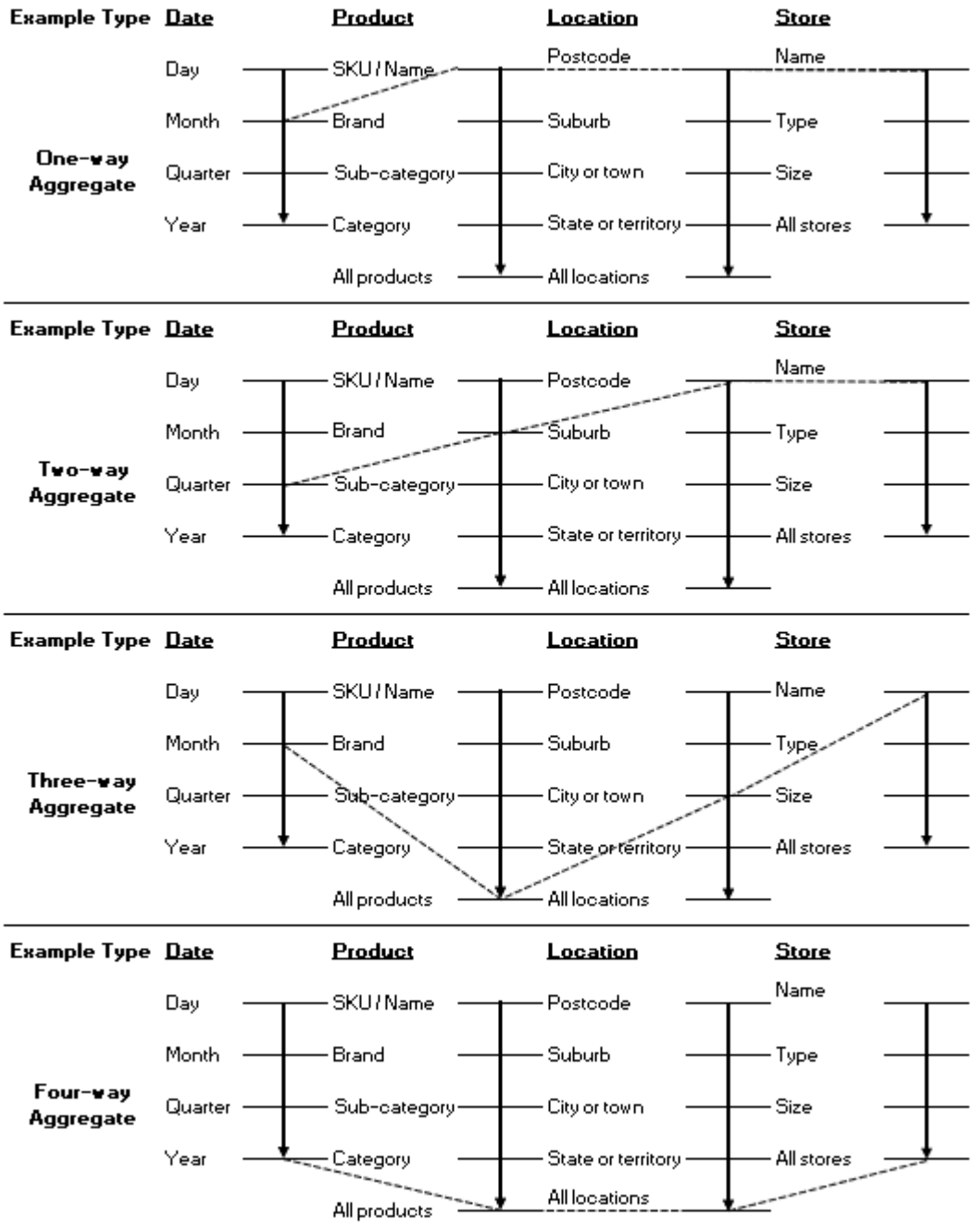


Figure 6: Examples of one- to four-way aggregate tables

Step Seven: A Lattice of Cuboids

Part A

Northwind Equestrian's six-dimensional data can be displayed as a series of five-dimensional data cubes. Furthermore, a cuboid can be generated for each of the possible subsets of the six dimensions. The result is a lattice of cuboids (figure 7); each of which shows the data at a different level of summarisation or 'group-by' (Han, Kamber, & Pei, 2011). At the top is the apex cuboid, which summarises the data over all six dimensions, and at the bottom is the base cuboid, which contains no dimension summarisation.

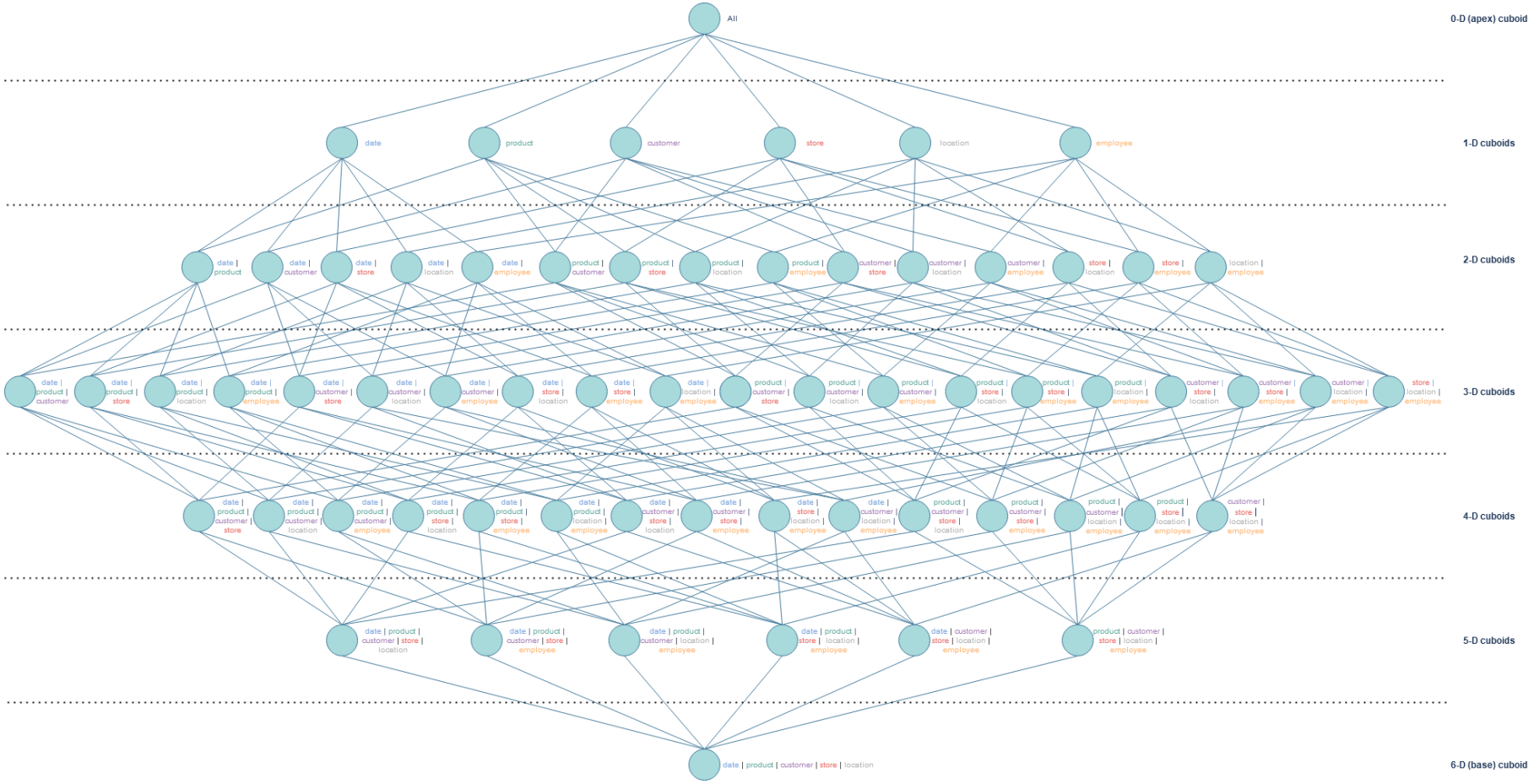


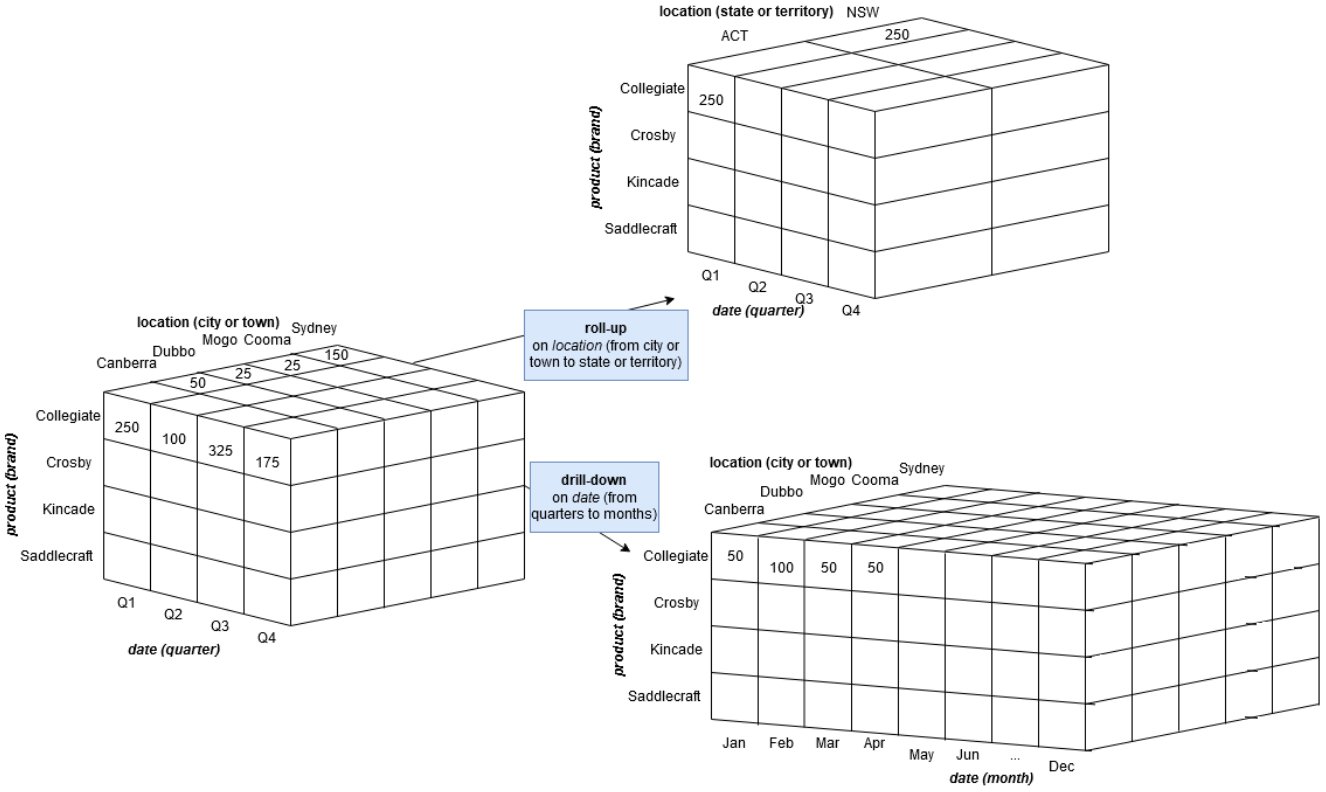
Figure 7: Lattice of cuboids

Part B

Roll-up and drill-down are two common OLAP operations performed on multi-dimensional data. Roll-up performs aggregation on a data cube by either climbing up the concept hierarchy of a dimension or by performing dimension reduction (Han et al., 2011). An example would be to ascend the location concept hierarchy from city or town to state or territory. Alternatively, if the sales data is being aggregated by both product brand and location, then dimension reduction would mean removing one dimension and aggregating by only brand or location.

Drill-down is the opposite of roll-up, which means that it increases the granularity of the data. Drill-down can be achieved by either stepping-down through a dimension's concept hierarchy or by introducing additional dimensions (Han et al., 2011). An example would be to descend the date concept hierarchy from quarter to month. Alternatively, if the sales data is being aggregated only by date, then drill-down would mean analysing the data by both the date and location dimensions.

Examples of drill-down and roll-up are shown in figure 8.



Step Eight: Data Cube Computation

Question A

The number of cuboids for a *n*-dimensional cuboid with no hierarchies is the same as the number of non-empty cuboids in a full data cube, which is equal to $2^n = 2^{10} = 1024$.

Question B

Any cell from a non-base cuboid is an aggregate cell, which means that it aggregates over one or more dimensions (Han, Kamber, & Pei, 2011). If each base cell generates $2^n - 1$ non-empty aggregate cells, then there will be in total $(2^n - 1) \times 3 = 3 \times 2^{10} - 3$ non-empty aggregate cells given the existence of three base cells.

However, four of these aggregate cells overlap:

- $(*, *, *, d_4, ..., d_9, d_{10})$ is common to all three base cells
- $(*, *, d_3, d_4, ..., d_9, d_{10})$ is common to base cells one and two
- $(*, d_2, *, d_4, ..., d_9, d_{10})$ is common to base cells one and three, and
- $(d_1, *, *, d_4, ..., d_9, d_{10})$ is common to base cells two and three.

Since this indicates that there are only seven unique values in the cuboid and hence 5×2^7 overlapping aggregate cells, the total number of non-empty aggregate cells in the full cube is $(3 \times 2^{10} - 3) - (5 \times 2^7) = 2429$.

Question C

Iceberg cubes are partially materialised cubes (Han et al., 2011), which means that only cells containing a measure value above a specified threshold (the minimum support threshold) are materialised.

Each of the four overlapping aggregate cells from the previous question have a count greater than equal to two ($(*, *, d_3, d_4, ..., d_9, d_{10})$, $(*, d_2, *, d_4, ..., d_9, d_{10})$ and $(d_1, *, *, d_4, ..., d_9, d_{10})$ all have a count of two, and $(*, *, *, d_4, ..., d_9, d_{10})$ has a count of three). Therefore, the total number of non-empty aggregate cells in an iceberg cube where $count \geq 2$ is $4 \times 2^7 = 512$.

Question D

There are seven closed cells in the full cube. These are:

- the three base cells $(a_1, d_2, d_3, d_4, ..., d_9, d_{10})$, $(d_1, b_2, d_3, d_4, ..., d_9, d_{10})$ and $(d_1, d_2, c_3, d_4, ..., d_9, d_{10})$, and
- the four overlapping aggregate cells $(*, *, d_3, d_4, ..., d_9, d_{10})$, $(*, d_2, *, d_4, ..., d_9, d_{10})$, $(d_1, *, *, d_4, ..., d_9, d_{10})$ and $(*, *, *, d_4, ..., d_9, d_{10})$.

Total word count: 1,796 words

Reference List

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: Concepts and techniques*. Saint Louis, USA: Elsevier Science & Technology.

Kimball, R., & Ross, M. (2013). *The data warehouse toolkit: The definitive guide to dimensional modeling* (3rd ed.). Hoboken, N.J.: Wiley.

Ponniah, P. (2010). *Data warehousing fundamentals for IT professionals* (2nd ed.). Hoboken, N.J.: Wiley.