⇒ Properties of Fact Table

① Concatenated key

- A row in a fact table refers to a combination of rows from all dimension tables
- this row is identified by primary keys of all dimension tables, so the pk is a concat of all DT pk's

② Data Grain

- level of detail for metrics
- we may choose for a transaction level or some aggregated higher level.

③ Fully Additive Measures

- attributes whose values can be summed up by simple addition for aggregation
- example: cost of product, quantity ordered

④ Semi-additive Measures

- often derived from other attributes & cannot be directly added
- example: profit percentage

⑤ Table Deep, Not Wide

- typically fewer attr. than dimension table
- Large number of records

Scanned by CamScanner

⑥ Sparse Data

- there are combinations of dimension table attributes, for which fact table row will have null values, ex: holidays have no deliveries
- such data can be omitted from the fact table

⑦ Degenerate Dimensions

- some attributes (like order no., invoice number etc) are neither facts nor strictly dimensional
- yet they are useful for some analysis.
  ↳ ex: avg no. of products per order

⇒ Factless Fact Table

- apart from the concatenated key, the fact table contains facts or measures.
- However, there are business events or coverage that can be represented in the fact table, although no measures are associated with it.
- Ex: For attendance we need not store 1 always, the presence of record in fact table indicates occurance of event.

⇒ Data Granularity

- it represents the level of detail in the fact table
- Keeping fact table at lowest level allows user to drill down to detail of operational system.
- fact tables must at a level corresponding to the dimension tables, as it allows changes like adding a new dimension attributes.

Scanned by CamScanner

- however lower level, means high cost of storage and maintenance
- advantages of granular fact tables:
  (i) easy adding of dimensions & attributes
  (ii) serve as natural destination for data from current ops system
  (iii) Can feed data in data mining applications

✡ Star Schema keys

(1) Primary keys

- each row in dimension table is uniquely identified by a primary key.
- it is not advised to use operational system unique key say product code in DW as they may get reassigned and then contlict with historical data

(2) Surrogate keys

- principles applied when choosing pk for DT:
  (i) avoid built-in meaning in the pk
  (ii) do not use prod. sys keys as pk
- surrogate keys are simple sys. generated sequence numbers
- they are mapped to prod. sys. keys
- store prod. keys as attributes

Scanned by CamScanner

③ Foreign keys

- each DT is 1-to-1 mapped with central FT.
- so pk of each DT is FK in the FT

⇒ Choosing pk for fact tables

① Single compound pk

drawback
- length = total length of keys from DT's
- both the compound, &, individual keys from DT need to be stored, so increases size of FT

Best option

② Concatenated pk

- pk = concat of pk from each DT
- no need to store pk from DT as fk
- individual parts of pk serve as fk

③ Generated PK

- independent of keys from DT
- same drawback as ① i.e unecessary increase in size.

☆ Advantages of STAR schema

① Easy for Users to Understand

- When users form queries, they ~~need~~ should know what to ask for
- they need to comprehend relationships
- STAR schema thinks alike i.e in terms of significant business metrics and dimensions
- It defines the join paths exactly the same way users normally visualize the relationship
- easy to use vehicle for communicating with users during development

Scanned by CamScanner

(2) Optimizes Navigation

- simple navigation for complex queries
- metrics are kept in middle as fact table, different dimension attribute in DT
- we can move quickly to analyse.

(3) Most Suitable for Query Processing

- query centric structure
- irrespective of number of participating dimensions in a query (may be complex), the query is executed as
  (a) select row from DT using query param filters
  (b) find corresponding FT rows
- provides ability to drill down and roll up.

(4) STAR join & STAR index

- allows query processer to use better execution plans.
- STAR join: a high speed, single pass, parallelizable multitable join. allows to join more than 2 tables in a single operation
- STAR index: specialized index to accelerate join performance. speed up joins b/w DT & FT

Scanned by CamScanner

✡ Dimensional Modelling : Advanced Topics

✡ Updates to Dimension Tables

- Compared to FT, DT are more stable and less volatile
- however unlike fact table apart from increase in rows, increase in attr. also takes place.

⇒ Changes in DT

Slowly Changing Dimensions

- most dimensions are generally const. over time
- many change slowly
- product key of source record does not change
- desc. & others change slowly
- Overwriting is avoided in DW
- need to look into what info. is to be presereed

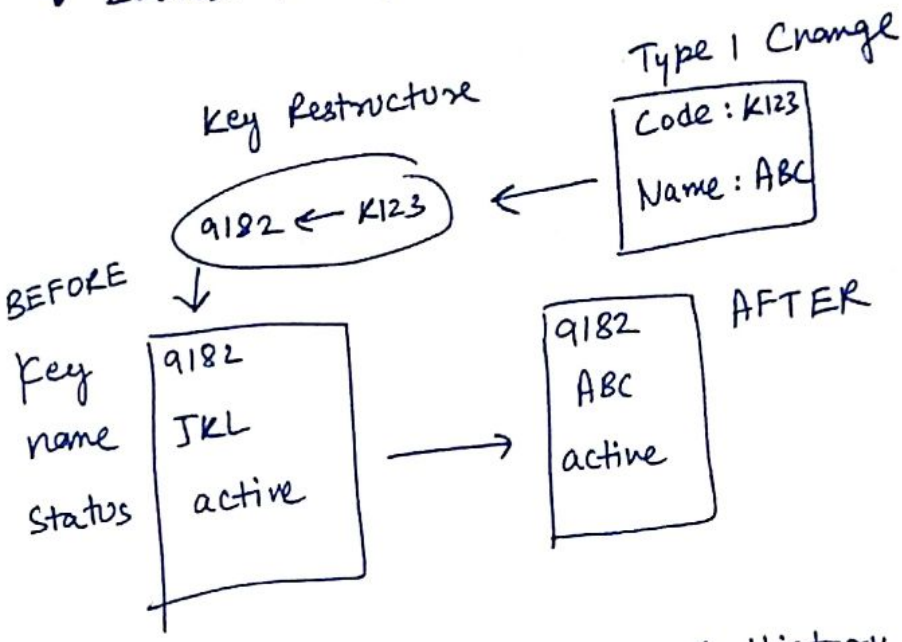
⇒ Types of changes in DT

① Type 1 Changes : Correction of errors

- Nature
  - usually relate to correction of errors in source
  - eg: spelling errors, status changes
  - mostly no need to preserne old values, so they can be discarded in favour of new values.

Scanned by CamScanner

- Applying:
  - Overwrite attr value in DT row with new value
  - Discard old value
  - Easiest to implement, no changes to keys

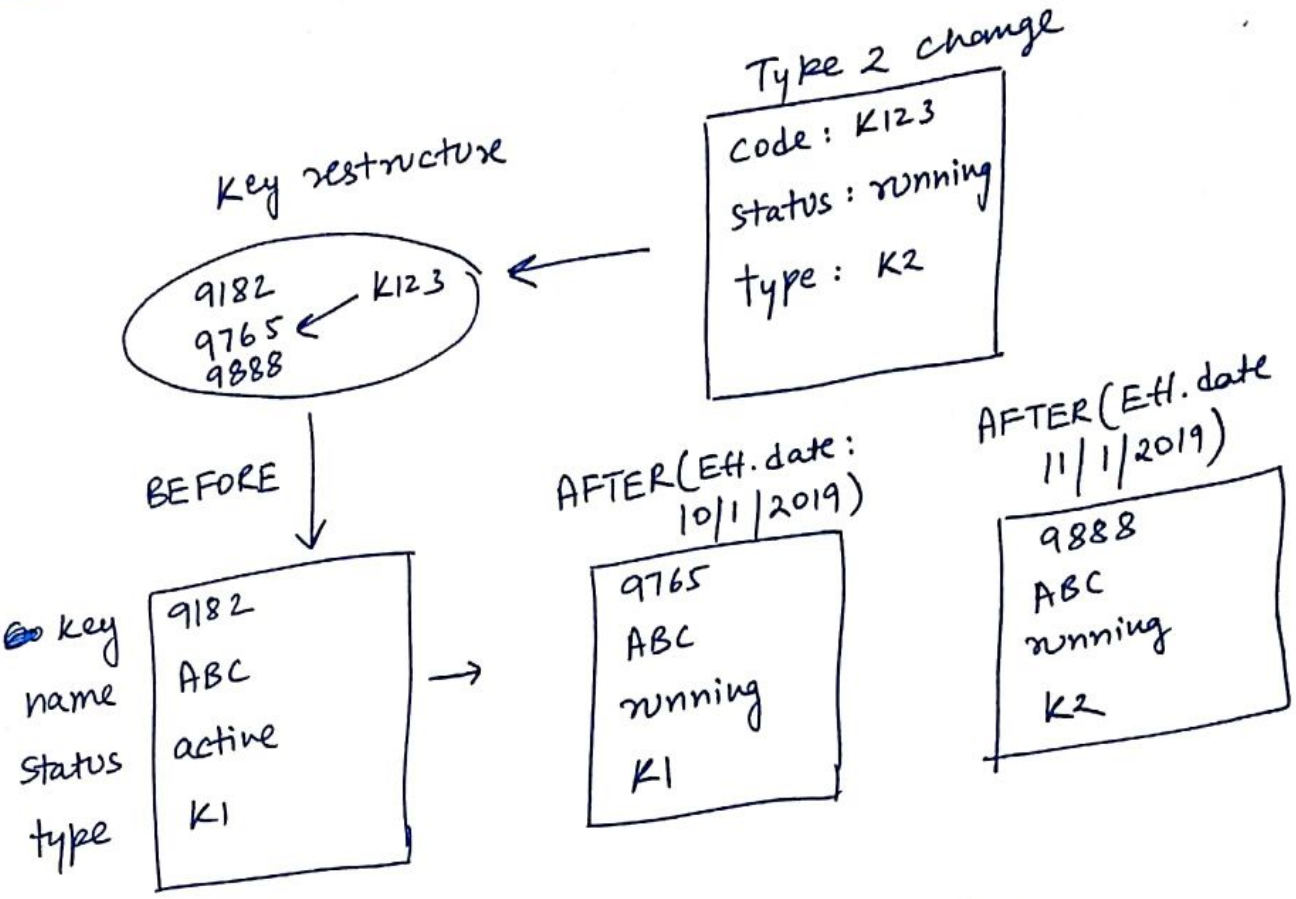


(2) Type 2 Changes: Preservation of History

- Nature:
  - we may be required to track changes
  - ~~need to change~~ usually relate to true changes in ops. systems
  - need to preserve history
  - partitions the history in DW
  - every change for the same attribute must be preserved

- Applying:
  - add a new DT row, with new value of attr
  - effective date field may be included in DT
  - no changes to original row in DT
  - Key of original row is unaffected
  - new row has a new surrogate key

Scanned by CamScanner

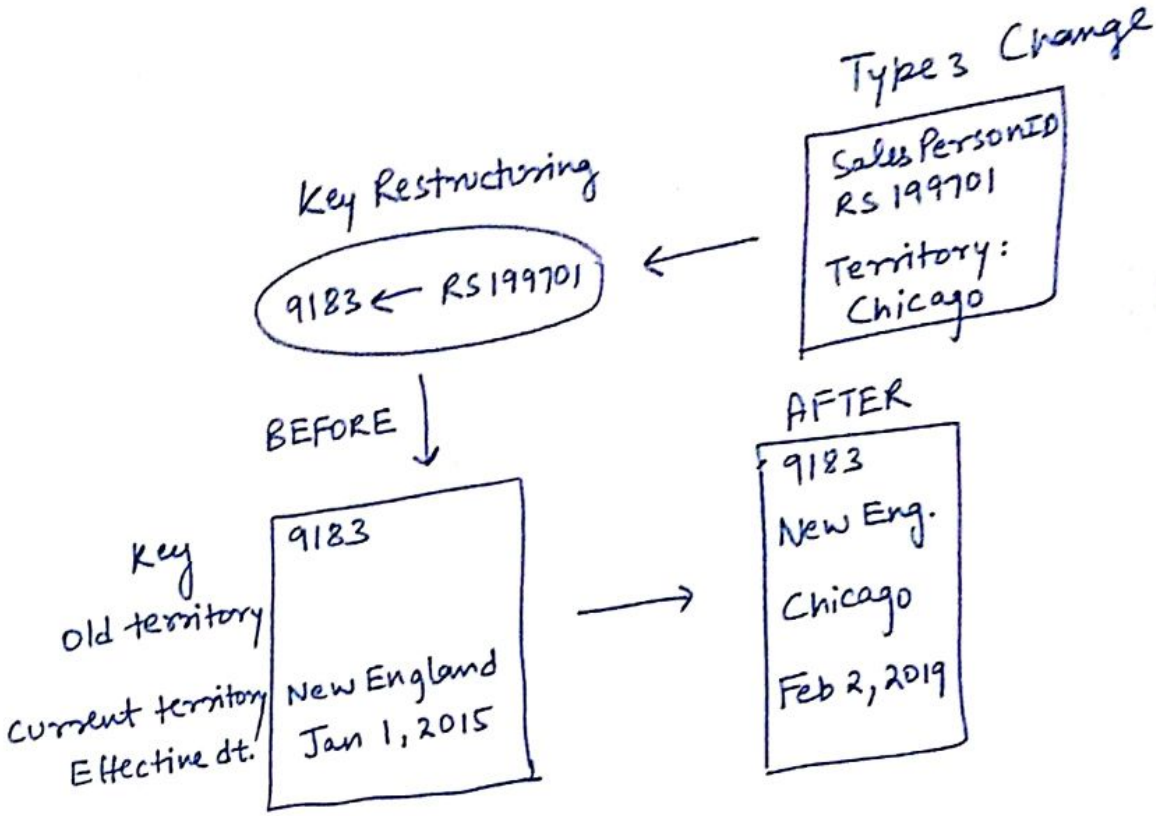


③ Type 3 Changes: Tentative Soft Revisions

- Nature:
  - relate to soft changes, tentative
  - need to track history of old & new values
  - used to compare performance across transition
  - provide ability to track forward & backward.

- Applying
  - Add an "old" field in DT for affected attr
  - Push existing value from current to old
  - keep new value in current field
  - add an effective date
  - key is not affected
  - no new row is added
  - queries revise to use old & new values

Scanned by CamScanner



✡ Large Dimensions

- may be very deep → have many rows
- may be very wide → have many attributes
- need special considerations
- In a DW, customer & product dimensions are typically large.

(a) ~~Product~~ Customer Dimension

- about 10 million rows
- 150 attributes average
- can have multiple hierarchies

(b) Product Dimension

- 100,000 product variations
- about 100 attributes
- can have multiple hierarchies

Scanned by CamScanner

- Issues that need optimization in Large dim:

  - population of tables
  - browse performance for unconstrained
  - browsing time for cross constrained
  - inefficiencies in fact table queries
  - additional row in Type 2 changes

✡ Rapidly Changing Dimensions

- When dimensions are large, Type 2 changes create large rows in numbers when changes happen rapidly
- Effective way is to break the dimension into simpler dimension tables
- So rapidly changing parts separate out.
- Example: custome → customer + behaviour
  (rapid) (normal) (rapid)

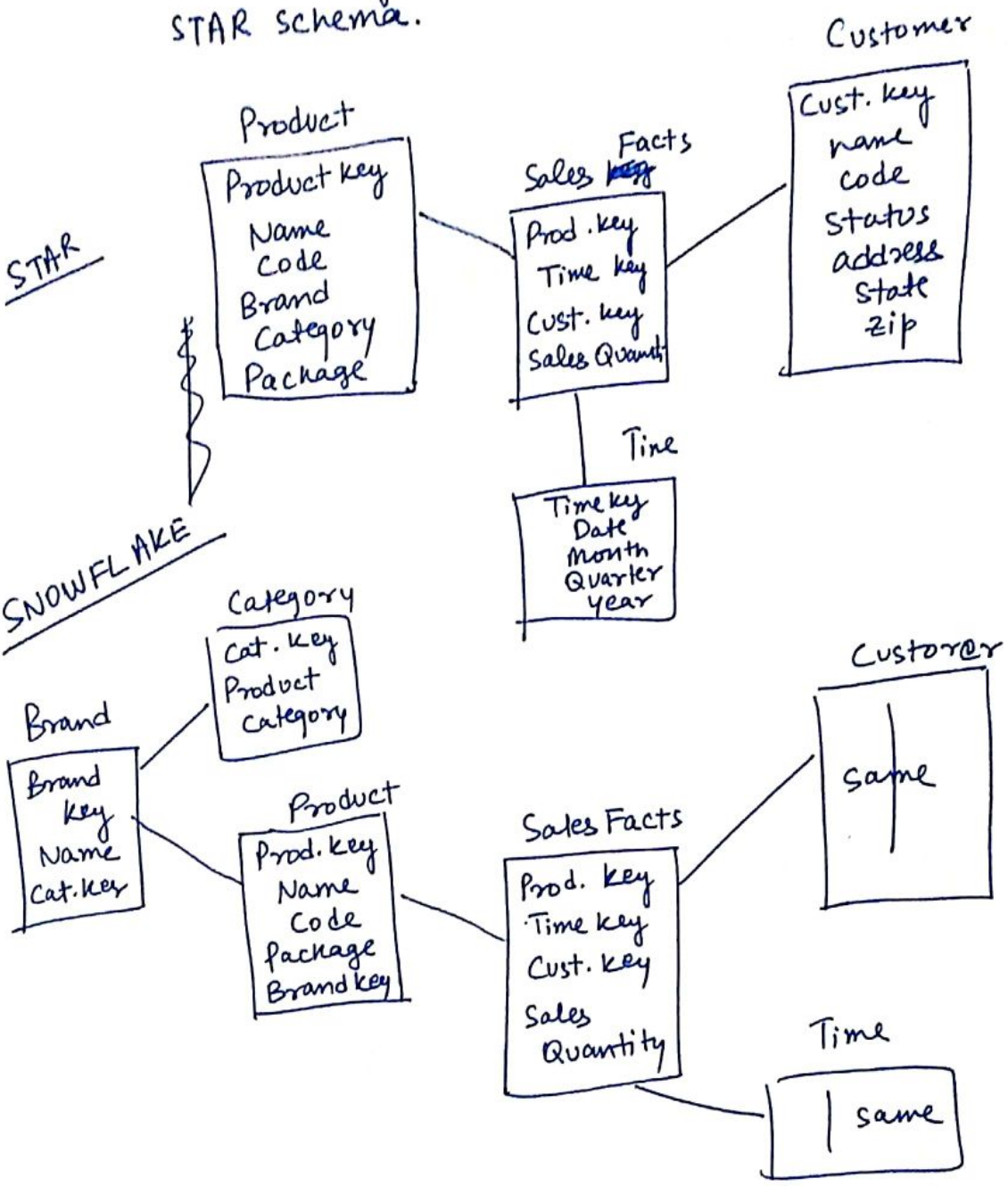
✡ Junk Dimensions

- Some fields like yes/no flags, textual codes etc. are too obscure to be of real value, hence left out.
- They may not be included as significant fields in DT
- How to deal with them [without discarding]
  (i) Place them in fact table. But swell up FT.
  (ii) Make each into its own dimension. But ↑ useless dim
  (iii) Keep only useful fields. For rest group them into a single junk dimension
- Junk dimension attributes are useful for constraining queries based on flag / text values.

Scanned by CamScanner

✡ The Snow Flake Schema

- Snowflaking is a method of normalizing the STAR schema.



Scanned by CamScanner

need

- Assume we have 500,000 product dimension rows where there are 500 brands which fall under 10 categories.
  If product dimension table is not indexed on category, any query would run through 500,000 rows
  Now it it is normalized to separate out brand & category it only goes through 10 rows

- Way for normalizing DT:
  - Partially normalize only a few dimensions
  - Fully normalize a few, partially few
  - Partially normalize all
  - Fully normalize all

- In snowflake schema, the attributes with low cardinality are removed to form separate DT's. New tables link to original using artificial keys.

⇒ Advantages

- Small savings in storage space.
  Say we remove 20 byte category names from 500,000 rows. To do this we add a 4 byte artificial key. So save 16 * 500,000 bytes ≈ 8 MB
  On any data is 200 MB so. just 4%.
- Normalized structures are easier to update & maintain

Scanned by CamScanner

⇒ Disadvantages

- Schema becomes less intuitive for end user
- Ability to browse through contents becomes difficult
- Degraded query performance due to joins

⇒ When to Snowflake

- Some situations require to form subdimensions
  example: separating demographic details from customer [like city population, pollution index, quality of life]

✡ Aggregate Fact Tables

- Aggregates are precalculated summaries derived from the most granular FT.
- Typically FT are billions of rows.

Need for Aggregates:
- Queries require summaries, but need granularity for drilldown and other advantages.
- So if we have precalculated aggregates, the query performance drastically improves.
- Aggregates have fewer rows than base tables

- Can be: [Example: these exist 3 dim]
  ① One way: higher in only 1 dimensions
  ex: Territory by product by date
  ② Two way: higher in only 2 dimensions
  ③ Three way: higher in only 3 dimension

Scanned by CamScanner

DWDM

✡ Data Extraction, Transform & Loading

- ETL functions reshape the relevant data from the source systems into useful info. to be stored in DW

✡ ETL Overview

- Part of data aquisation and data storage components
- backend processes that:
  - cover extraction of data from sources
  - include all func. & procedures to transform data into required formats & structures
  - include fn for physically moving data into the DW repository.

- Problems faced:
  - diverse source systems
  - sources on multiple platforms
  - sometimes obslete & legacy sources
  - historical data is not preserved in ops systems
  - low quality of data
  - frequent changes in source systems
  - inconsistency among sources
  - different format across sources
- Accounts for 50-70% of project effort.
- Time consuming and ardous to identify data in each source (due to diversity) and then load massive volumes of data

Scanned by CamScanner

- Major Steps in ETL :

(i) Determine all target data needed in DW
(ii) Determine internal & external data sources
(iii) Prepare mapping from source to target
(iv) Establish comprehensive data extraction rules
(v) Determine trans. & cleansing rules
(vi) Plan for aggregate tables
(vii) Org. data staging area & test tools
(viii) Write procedures for data loads
(ix) ETL for DT
(X) ETL for FT

- Key Factors

(i) Need to focus on diversity of sources. Need a complete inventory.
(ii) Mass refreshes (like init. loading) take long times, so need to accordingly schedule them.

✡ Data Extraction

Scanned by CamScanner

DWDM

✡ Introduction to Data Mining

- aka KDD or Knowledge discovery from data

✡ Why Data Mining

- vast amounts of data are collected daily.
- analyzing such data is important.
- explosion in data volume is due to computerization of society & fast development collection and storage tools
- Ex: Wal-mart has millions of transactions per week at thousand branches
- Power tools are needed to automatically uncover valuable information ~~to~~ and transform it to Org. knowledge
- Data mining turns a large collection of data into knowledge

⇒ Data Mining as Evolution of IT

- db and data management industry evolved into development of:
  - data collection & database creation
  - data management
  - advanced data analysis
- progression from primitive file processing systems to sophisticated databases.
- need to move away from the situation "Data rich but Information poor"

Scanned by CamScanner

✡ What is Data Mining

- Similar terms: Knowledge mining from data, knowledge extraction, data archaeology.

- Steps in knowledge discovery from data:
  - (i) Data cleaning: remove noise & inconsistencies
  - (ii) Data integration: combine data from multiple sources
  - (iii) Data selection: retrieval of relevant data for analysis
  - (iv) Data transformation: cosolidate into summary and aggregations
  - (V) Data Mining: intelligent methods are applied to extract data patterns
  - (vi) Pattern evaluation: identify truly interesting patterns
  - (vii) Knowledge presentation: Use viz. to show mined knowledge to users.

- Broadly
  - Data mining is the process of discovering interesting patterns and knowledge from large amounts of data
  - The data sources include databases, data warehouses, the Web, or streaming data.

Scanned by CamScanner

✡ What Kind of Data Can be Mined

- data mining can be applied to any kind of data as long as the data are meaningful for a target application.

① Database Data

- consists of interrelated data
- a RDBMS, is a collection of tables, each table has some attributes & some tuples
- can be accessed by database queries (like SQL)
- mining is done by searching for trends or data patterns
- ex: analyse customer data to predict credit risk
- detect deviations from historical benchmarks
- most common, and richest info. repositories

② Data Warehouses

- A DW is a repository of information collected from multiple sources, stored at a single site in a unified schema.
- Constructed using ETL processes.
- Data is organized around business dimensions and required metrics
- Provides ability to drill down and roll up along multiple dimensions
- Provides inherent support for OLAP

Scanned by CamScanner

③ Transactional Data

- transaction can be customer's purchase, flight booking, user clicks
- Mostly mining is done to answer "which items exist in query set together".
- Market basket data analysis

④ Other kinds

- time series data : ex: stock exchange data
- streaming data : ex: sensors, CCTV
- Spatial data : ex: maps
- graph & network data : social networks

✡ What kinds of Applications are Targeted?

⇒ Business Intelligence

- critical for business to acquire better understanding of commercial context.
- BI provides historical, current and predictive views of business operations.
- Examples: reporting, OLAP, predictive analytics
- OLAP tools rely on DW
- Classification & prediction allow market analysis, sales prediction etc.
- Clustering helps in user segmentation.

Scanned by CamScanner

=> Web Search Engines

- specialized computer server that searches information on the Web.
- return web pages, images or files according to the user's search query.
- essentially very large data mining applications
- various data mining techniques used:
  - Crawling
  - indexing
  - searching
- challenges faced:
  - handling huge and ever growing amount of data
  - need online data i.e realtime resolution
  - most queries are used 1 times, need to be context aware.

Scanned by CamScanner

DWDM

✡ Data Objects and Attribute Types

- Data sets comprise of data objects
- A data obj. represents an entity
- They are described by attributes.

⇒ Attributes

- data field representing a characteristic/feature of a data object.
- aka dimension, feature, variable
- example: customer ID, name, address
- Observed values of a given attr. are called observations
- A set of attributes used to define an object is the attribute or feature vector.
- If data distribution involves 1 attribute it is univariate else multivariate.

- Types

① Nominal

- relate to names
- values are symbols or names of things
- each value represents some kind of category, code or state ⟶ so aka categorical
- values have no meaningful ordering, no maths applies
- values aka enumerations.
- no meaning of mean & median
- mode is used

Scanned by CamScanner

② Binary

- Nominal attribute with only 2 categories 0 or 1
- aka Boolean if states are true and false
- eg: smoker, adult
- they are symmetric if both states have same weight. Ex: gender
- asymmetric if outcomes are not equally important ex: more value associated with HIV tve

③ Ordinal

- possible values have meaningful order and a ranking among them.
- but magnitude of successive values is unknown.
- useful for subjective assessment of qualities
- often used in survey ratings:
  eg:→ very dissatisfied, dissatisfied, neutral, satisfied
- can be obtained by splitting value range into finite ordered sets
- mode and median are used
- mean is not defined
- qualitative in nature

Scanned by CamScanner

④ Numeric Attributes

- quantitative
- provide a measurable quantity
- Can be:

(a) Interval scaled

- scale is of equal size units
- values have order, can be +ve, 0, or -ve
- provide ranking
- quantify difference b/w values
- eg: temperature in °C or °F
- can compute mean, median, mode

(b) Ratio Scaled

- numeric with inherent zero point
- can be a multiple of another value
- values are ordered
- can compute mean, median, mode
- eg: temperature in K

⇒ Discrete v/s Continous Attributes

- ① Discrete
  - have a finite or countably infinite sets of value which may or may not be integers
  - ex: hair-color, smoker, color etc.

② Continous

- non discrete
- typically floating type
- eg: blood pressure

Scanned by CamScanner

✡ Statistical Descriptions of Data

⇒ Measuring Central Tendency

① mean

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N}$$

Weighted

$$\bar{x} = \frac{\sum_{i=1}^{N} w_i x_i}{\sum_{i=1}^{N} w_i}$$

- problem is sensitivity to extreme values

② Median

- middle value in an ordered set
- better for skewed data
- Can be approximated for ranges

$$\text{median} = L_1 + \left(\frac{N/2 - (\sum \text{freq})_l}{\text{freq}_{\text{median}}} \right) \text{width}$$

③ Mode

- ~~b~~ most freq. element in set
- unimodal, multimodal

④ Mid range

- avg (smallest, largest)

- Symmetric : mean = mode = median
- Positive Skew : Mean > median > mode
- Negative Skew : mode > median > mean

Scanned by CamScanner

⇒ Measuring Data Dispersion

① Range
- difference b/w max() and min()

② Quantiles
- points taken at regular intervals
- divide into equal sized consecutive sets.
- the k^{th} q-quantile is x if

$$\left(\frac{k}{q}\right)_{values} < x < \left(\frac{q-k}{q}\right) values$$

- 4 quantile → quartile (4 parts)
- IQR: Interquartile Range
  $= Q_3 - Q_1$

③ Five Number Summary
median (Q_2), Q_1, Q_3, min(), max()

④ Outliers
- atleast 1.5 * IQR above Q_3 or below Q_1

⑤ Boxplot
- used to viz. a dist.
- incorporate the 5 Number summary
  - ends are Q_1 and Q_3, length = IQR
  - median is a line within the box
  - 2 lines called whiskers mark the min() and max()

Scanned by CamScanner

(6) Variance and Std. Deviation

- low shows data is close to mean

$$Var = \sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2$$

✡ Measuring Data Similarity & Dissimilarity

- need to identify how alike or unalike items are
- a cluster is a collection of similar items
- outliers are dissimilar to general data points

⇒ Data Matrix

- aka object by attribute structure, 2 mode matrix
- stores n objects x p attributes
- each row corresponds to an object

⇒ Dissimilarity Matrix

- object - by- object structure, 1 mode matrix
- Stores a collection of proximities for n objects

$d(i,j)$ = dissimilarity b/w i and j

$d(i,i) = 0$ and & $d(i,j) = d(j,i)$

$sim(i,j) = 1 - d(i,j)$

Scanned by CamScanner

⇒ Proximity Measures for Nominal Attr.

$$d(i,j) = \frac{p - m}{p}$$

p = total number of attr. describing obj
m = total no. of matching values

⇒ PM for Binary Attr

$$d(i,j) = \frac{r+s}{q+r+s+t}$$ } when symmetric

Number

		j = 1	j = 0
i	1	q	r
	0	s	t

$$d(i,j) = \frac{r+s}{q+r+s}$$ } when asymmetric

$$sim(i,j) = 1 - d(i,j) = \frac{q}{q+r+s}$$ } Jaccard Coefficient

⇒ Dissimilarity in Numeric Data : Minkowski Distance

① Euclidean

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 - - -}$$

② Manhattan

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| - - - - - -$$

Scanned by CamScanner

- Conditions for being a metric

Satisfied by Mahattan & Euclidean:

(i) Non-Negativity : $d(i,j) \geq 0$

(ii) Identity of indiscenibles : $d(i,i) = 0$

(iii) Symmetry : $d(i,j) = d(j,i)$

(iv) $\triangle$ inequality : $d(i,j) \leq d(i,k) + d(k,j)$

(3) Minkowski

$$d(i,j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + \dots}$$

- generalisation

(4) Supremum or L_∞ or Chebyshev

- $h \to \infty$ in Minkowski

$$d(i,j) = \lim_{h \to \infty} \left(\sum_{f=1}^{p} |x_{if} - x_{jf}|^h \right) = \max_{f}^{p} |x_{if} - x_{jf}|$$

⇒ PM for Ordinal Attr

- first normalize to $[0, 1]$ range
- then apply any Minkowski Distance

⇒ Cosine Similarity

- used in document vectors

$$sim(x,y) = \frac{x \cdot y}{|x| \cdot |y|}$$

Scanned by CamScanner