Data Mining: Homework 1

Illusionna 2025XXXXXXXXX04 Artificial Intelligence School

19:06, Wednesday 5th November, 2025

Contents

1 Data Warehouse 1

(a) Star schema diagram 1

(b) OLAP operations 1

(c) Bitmap indexing 1

2 Hospital Test Data 2

(a) Statistics 3

(b) Boxplot 3

(c) Scatter plot 3

(d) Normalization 4

(e) Pearson coefficient 4

(f) Smooth data by mean 4

(g) Smooth data by boundary 5

3 Design data warehouse 5

1 Data Warehouse

Suppose that a data warehouse consists of four dimensions, date, spectator, location, and game, and two measures, count and charge, where charge is the fare that a spectator pays when watching a game on a given date. Spectators may be students, adults, or seniors, with each category having its own charge rate.

(a) Star schema diagram

Draw a star schema diagram for the data warehouse.

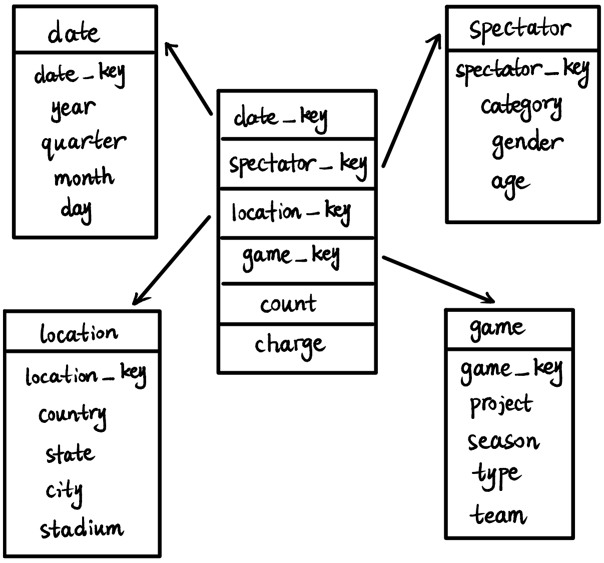

Figure 1: Star Schema Diagram

(b) OLAP operations

Starting with the base cuboid [date, spectator, location, game], what specific OLAP operations should one perform in order to list the total charge paid by spectators in Chicago in 1999?

```
slice for location = "Chicago"
roll-up on location from "stadium" to "city"
slice for date = "1999"
roll-up on date from "day" to "year"
roll-up on spectator from "all"
roll-up on game from "all"
```

(c) Bitmap indexing

Bitmap indexing is a very useful optimization technique. Please present the pros and cons of using bitmap indexing in this given data warehouse.

Pros

- Compared with traditional indexing structure, the bitmap indexing can quickly select multi-dimensional conditions, such as:

$$(\texttt{location.city} = \texttt{Chicago}) \bigoplus (\texttt{spectator.category} = \texttt{Adult})$$

- Very suitable for low cardinality domains, such as `spectator.category`.
- The storage space of RAM and Disk is very small when the dimension is low.
- It has an advantage in read-only or read-mostly OLAP like data warehouse.

Cons

- It is inefficient for high cardinality domains such as `date or game`, leading to the storage of RAM and Disk is large because of the sparse bitmap indexing structure.
- High maintenance cost when date warehouse is frequently updated.
- The bitmap indexing structure is not suitable for real-time system like cuboid [date, spectator, location, game].

2 Hospital Test Data

Suppose a hospital tested the age and body fat data for 18 random selected adults with the following result:

Table 1: Age and body fat data

age	%fat
23	9.59
23	26.5
27	7.8
27	17.8
39	31.4
41	25.9
47	27.4
49	27.2
50	31.2
52	34.6
54	42.5
54	28.8
56	33.4
57	30.2
58	34.1
58	32.9
60	41.2
61	35.7

(a) Statistics

Calculate the mean, median, and standard deviation of age and %fat.

$$\text{mean}_{\text{age}} = \frac{1}{n}\sum_{i=1}^{n} \text{age}_i \approx 46.44$$

$$\text{mean}_{\text{fat}} = \frac{1}{n}\sum_{i=1}^{n} \text{fat}_i \approx 28.78\%$$

$$\text{median}_{\text{age}} = 51$$

$$\text{median}_{\text{fat}} = 30.7\%$$

$$\text{standard}_{\text{age}} = \sqrt{\frac{\sum_{i=1}^{n} (\text{age}_i - \overline{\text{age}})^2}{n-1}} \approx 13.22$$

$$\text{standard}_{\text{fat}} = \sqrt{\frac{\sum_{i=1}^{n} \left(\text{fat}_i - \overline{\text{fat}}\right)^2}{n-1}} \approx 9.25\%$$

(b) Boxplot

Draw the boxplots for age and %fat.

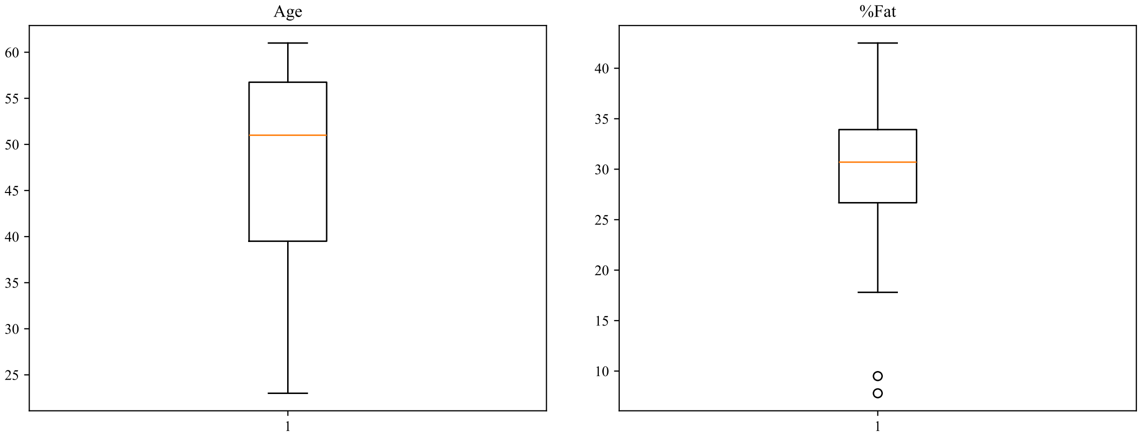

Figure 2: Boxplot

(c) Scatter plot

Draw a scatter plot based on these two variables.

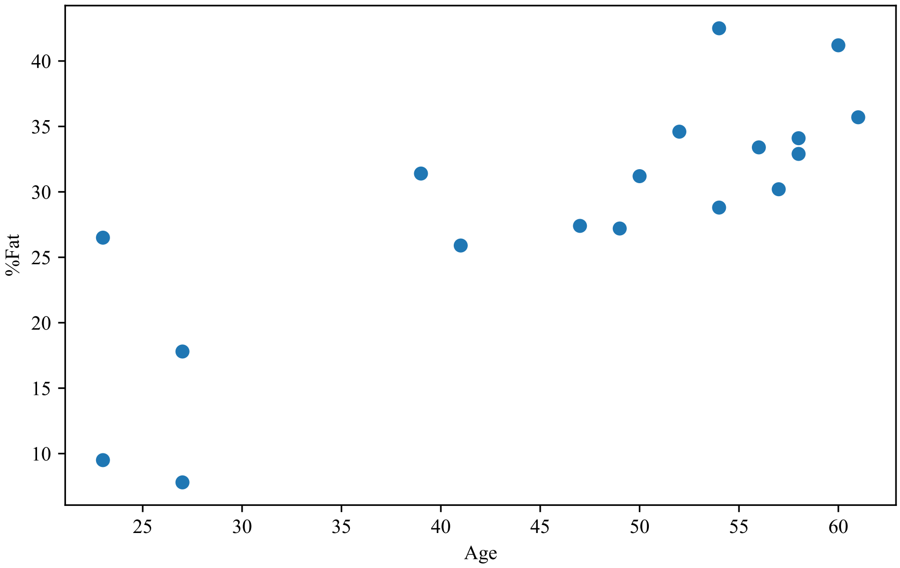

Figure 3: Scatter Diagram

(d) Normalization

Normalize age based on min-max normalization.

$$f(x) = \frac{x - \min(x)}{\max(x) - \min(x)}$$

```
normalizede_age = [
    0, 0, 0.105, 0.105, 0.421, 0.474, 0.632, 0.684, 0.711,
    0.763, 0.816, 0.816, 0.868, 0.895, 0.921, 0.921, 0.974, 1
]
```

(e) Pearson coefficient

Calculate the correlation coefficient (Pearson's product moment coefficient). Are these two variables positively or negatively correlated?

$$\rho(\text{age},\ \text{fat}) = \frac{\sum_{i=1}^{n}(\text{age}_i - \overline{\text{age}})(\text{fat}_i - \overline{\text{fat}})}{(n-1)\sigma_{\text{age}}\sigma_{\text{fat}}} \approx 0.8176$$

The variable `age` is positively correlated with the variable `fat`.

(f) Smooth data by mean

Smooth the fat data by bin means, using a bin depth of 6.

```
Partition using equal frequency approach:
    - Bin 1: (7.8, 9.5, 17.8, 25.9, 26.5, 27.2)
    - Bin 2: (27.4, 28.8, 30.2, 31.2, 31.4, 32.9)
    - Bin 3: (33.4, 34.1, 34.6, 35.7, 41.2, 42.5)
```

```
Smoothing by bin means:
    - Bin 1: (19.12, 19.12, 19.12, 19.12, 19.12, 19.12)
    - Bin 2: (30.32, 30.32, 30.32, 30.32, 30.32, 30.32)
    - Bin 3: (36.92, 36.92, 36.92, 36.92, 36.92, 36.92)
```

(g) Smooth data by boundary

Smooth the fat data by bin boundaries, using a bin depth of 6.

```
Smoothing by bin boundaries:
    - Bin 1: (7.8, 7.8, 27.2, 27.2, 27.2, 27.2)
    - Bin 2: (27.4, 27.4, 32.9, 32.9, 32.9, 32.9)
    - Bin 3: (33.4, 33.4, 33.4, 33.4, 42.5, 42.5)
```

3 Design data warehouse

Design a data warehouse for a regional weather bureau. The weather bureau has about 1000 probes, which are scattered throughout various land and ocean locations in the region to collect basic weather data, including air pressure, temperature, and precipitation at each hour. All data are sent to the central station, which has collected such data for over 10 years. Your design should facilitate efficient querying and online analytical processing, and derive general weather patterns in multidimensional space. (Note: please present the fact table(s) and the dimension tables with concept hierarchy in star schema, snowflake schema or galaxy schema)

Table 2: Dimension table

time	region	probe	measure
hour	type	category	temperature
day	coordinate	manufacturer	air pressure
month	country	model	precipitation
quarter	state		wind speed
year	city		
	name		

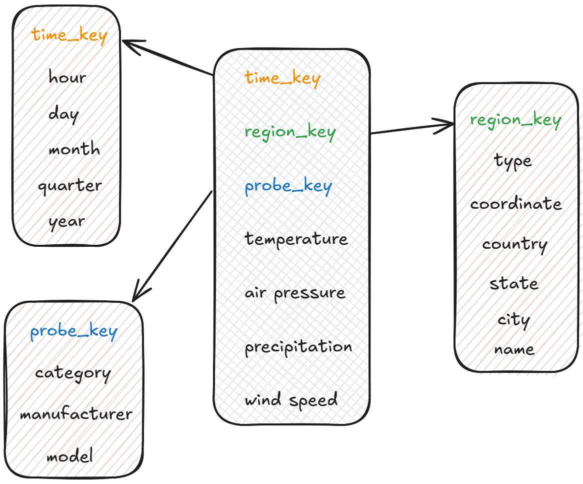

Figure 4: Star Schema Diagram

Cube Definition Syntax in DMQL:

```
-- Cube Definition (Fact Table)
define cube weather_bureau [time, region, probe]: [temperature, air pressure, precipitation, wind speed]

-- Dimension Definition (Dimension Table)
define dimension time_table as [hour, day, month, quarter, year]
define dimension region_table as [type, coordinate, country, state, city, name]
define dimension probe_table as [category, manufacturer, model]

-- Special Case (Shared Dimension Table)
define dimension time_key as time_table.time_key in cube weather_bureau.time_key
define dimension region_key as region_table.region_key in cube weather_bureau.region_key
define dimension probe_key as probe_table.probe_key in cube weather_bureau.probe_key
```

Defining Star Schema in DMQL:

```
define cube weather_bureau [time_key, region_key, probe_key]: temperature, air pressure, precipitation, wind speed
define dimension time as (time_key, hour, day, month, quarter, year)
define dimension region as (region_key, type, coordinate, country, state, city, name)
define dimension probe as (probe_key, category, manufacturer, model)
```

Since the weather bureau has 1000 probes scattered throughout various land and ocean locations, it need to construct a spatial data warehouse so that we can view weather patterns (like temperature, air pressure, precipitation and wind speed etc.) on a map by time, by region and by probe. We can dynamically drill-down, roll-up and slice along any dimension to explore certain patterns.

The OLAP operations (such as roll-up or slice) can be implemented in the data cube if a spatial data cube contains dimensions but not measures. If we need to use spatial measures in a data cube, we can selectively pre-computation some measures. The structure of cube selected for program depends on access frequency, access priority, online computation and so on.