Information Policy

Data Warehouse

D04. Multi-Dimensional Modeling

- Code: 164323-03
- Course: Information Policy
- Period: Spring 2013
- Professor: Sync Sangwon Lee, Ph. D

Contents

01. Multi-Dimensional Modeling

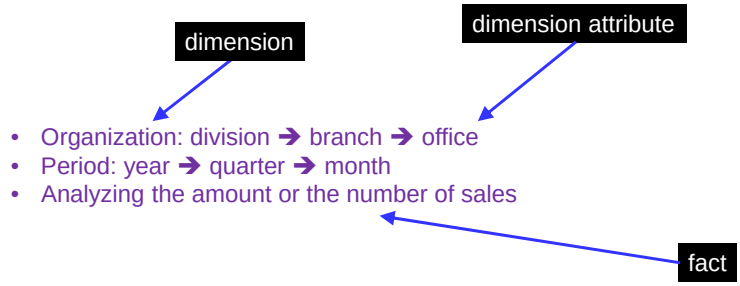
- Multi-Dimensional Modeling
  - Speedy response to complex queries which can accept user's various requirements for ROLAP solution
  - Cf. MOLAP
    - MDB is multi-dimensional itself. So an additional modeling is not needed.
  - Cf. ER modeling
    - It maintains integrity in the OLTP environment.

01. Multi-Dimensional Modeling

- Multi-Dimensional Modeling
  - Handling large data
    - For this, system tuning is needed to control parameters in RDB.
    - Various functions of OLAP are applied to multi-dimensional modeling.
    - The correctness of SQL of ROLAP is verified.

01. Multi-Dimensional Modeling

- Ex.

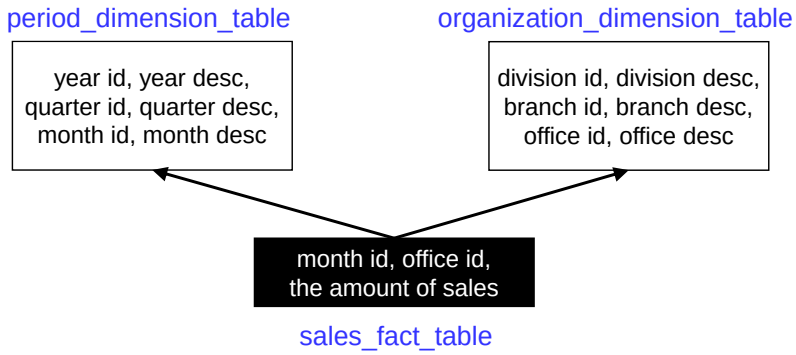


02. Star Schema vs. Snowflake Schema

- Star Schema
  - Denormalized snowflake schema
    - A technology to design summary tables
  - Characteristics
    - Assigning one dimension table to each dimension
    - Denormalized dimension tables + Fact table
      - Redundancy
      - Large-sized tables
  - Demerits
    - Unfit for use when a relationship b/t dimension attributes is N:N.

02. Star Schema vs. Snowflake Schema

- Star Schema
  - Ex.



02. Star Schema vs. Snowflake Schema

- Star Schema
  - Ex.

What is the amount of sales by month, branch?

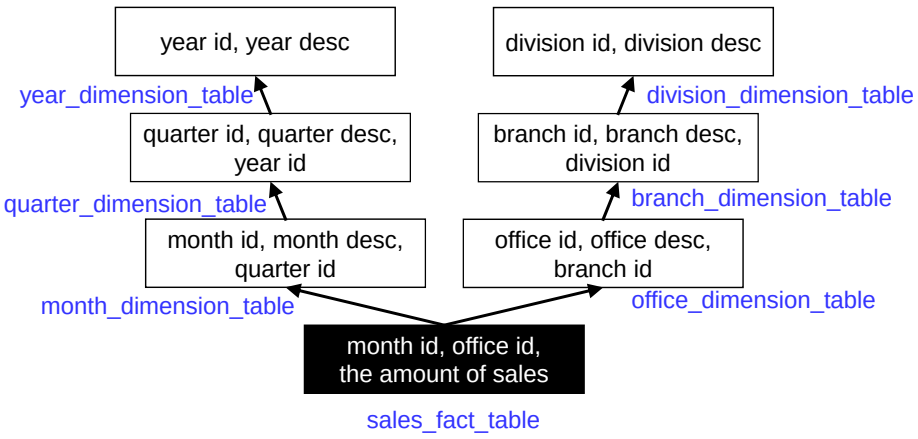
```
SELECT month_desc, branch_desc, sum(sales)
FROM   sales_table, period_table, organization_table
WHERE sales_table.month_id = period_table.month_id
AND      sales_table.office_id = organization_table.office_id
GROUP BY month_id, branch_id
```

02. Star Schema vs. Snowflake Schema

- Snowflake Schema
  - Normalized star schema
  - Characteristics
    - Assigning one dimension table to each dimension attribute
      - Dimension information + key for joining the upper dimension
    - Normalized dimension attribute tables + Fact table
      - No redundancy
      - Small-sized tables
  - Demerits
    - Many star joins are needed.
      - Star join: Many small tables join with one big table.

02. Star Schema vs. Snowflake Schema

- Snowflake Schema
  - Ex.



02. Star Schema vs. Snowflake Schema

- Snowflake Schema
  - Ex.

What is the amount of sales by month, branch?

```
SELECT month_desc, branch_desc, sum(sales)
FROM   sales_table, month_table, office_table, branch_table
WHERE sales_table.month_id = month_table.month_id
AND      sales_table.office_id = office_table.office_id
AND      office_table.branch_id = branch_table.branch_id
GROUP BY month_id, branch_id
```


02. Star Schema vs. Snowflake Schema

- Star Schema vs. Snowflake Schema

- Star Schema
  - Assigning one dimension table to each dimension
  - Denormalized dimension tables + Fact table
    - Redundancy
    - Large-sized tables

Denormalized ↑ ↓ **Normalized**

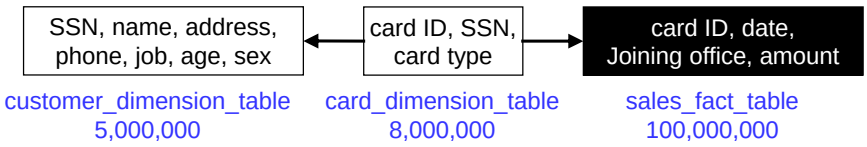
- Snowflake Schema
  - Assigning one dimension table to each dimension attribute
  - Normalized dimension attribute tables + Fact table
    - No redundancy
    - Small-sized tables

03. Huge Dimension

- Backgrounds of Huge Dimension
  - The size of tables is getting huger and huger.
    - ➔ Huge dimension
  - Both fact and dimension is possible.
  - Slow performance with many joins.
  - Solution
    - Bitmap index
    - B-tree index
    - Summary table

03. Huge Dimension

- Ex.
  - Customer, card, sales
    - ➔ Card is both dimension of sales and fact of customer.
  - What is the amount of sales of men in Gangnam, February, 2001?
    - ➔ Joins: 5,000,000 x 8,000,000 x 100,000,000

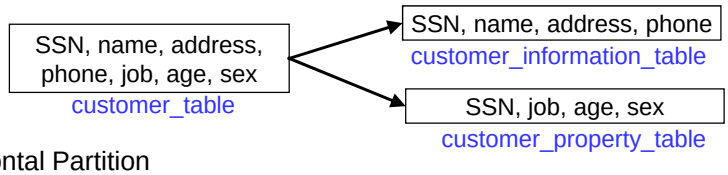


04. Partition

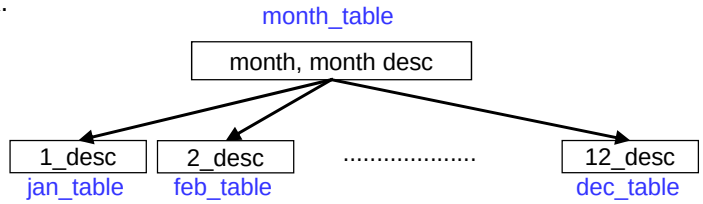
- Goal of Partition
  - High performance
- Types of Partition
  - Vertical partition
  - Horizontal partition

04. Partition

- Vertical Partition
  - Ex.



- Horizontal Partition
  - Ex.

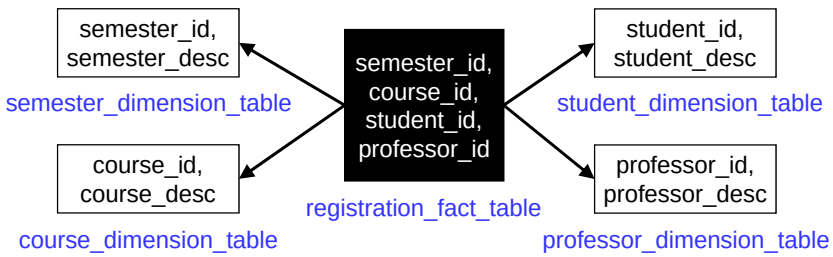


05. Fact Table without Fact

- Fact Table without Fact
  - There exist only ID values, no fact columns.
  - Used for counting

05. Fact Table without Fact

- Ex.
  - How many students by professors?
  - What is the course with the maximal members of students?



06. Bitmap Index

- Bitmap Index
  - Necessary for DW performance
  - Used when data are frequently updated
  - High performance with logic operations

06. Bitmap Index

- Ex.

ID	PNM	ID	PNM	ID	PNM	ID	PNM	ID	PNM
1	TV	2	VCR	3	CRM	4	AUD	5	TV
6	VCR	7	CRM	8	AUD	9	TV	10	VCR
11	CRM	12	AUD	13	TV	14	VCR	15	CRM
16	AUD	17	TV	18	VCR	19	CRM	20	AUD

<TV, rowid>: 10001000100010001000
<VCR, rowid>: 01000100010001000100
<CRM, rowid>: 00100010001000100010
<AUD, rowid>: 00010001000100010001

07. Meta Data

- Meta Data
  - Data of Data
  - The core of DW
  - Meta data do not change after operation data are loaded on DW.

07. Meta Data

- Structure