

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

Bộ môn Hệ Thống Thông Tin



BÁO CÁO

Chương 9: Star Schema Transformation

Giảng Viên: TS. Trần Việt Trung

Nhóm sinh viên:

Bùi Sĩ Tuấn
Nguyễn Đức Trọng
Lê Tuấn Anh
Đỗ Đăng Hiền

MSSV:

20112408
20112354
20111117
20111581

Hà Nội, ngày 03 tháng 05 năm 2015

MỤC LỤC

ĐẶT VẤN ĐỀ	3
TÀI LIỆU THAM KHẢO	4
I. Lược đồ trong mô hình kho dữ liệu	5
1. Lược đồ hình sao	5
2. Mô hình lược đồ hình bông tuyết	7
II. Star Transformation	8
1. Truy vấn trong lược đồ hình sao	8
2. Star Transformation là gì?	9
2.1. Bitmap index	9
2.2. Định nghĩa Star Transformation	10
2.3. Lưu ý	10
3. Quy trình thực hiện Star Transformation	11
3.1. Viết lại câu truy vấn	11
3.2. Pha thứ nhất – thu thập các bản ghi từ một bảng dimension	11
3.3. Pha thứ nhất – thu thập các bản ghi từ tất cả bảng dimension	13
3.4. Pha thứ hai - Kết nối tập kết quả trung gian với các bảng dimensions	13
4. Cải thiện hiệu suất cho “Star Transformation”	14
4.1. Sử dụng bảng trung gian	14
4.2. Sử dụng Bitmap Join Indexes	15
4.3. Các mô hình kết nối của Bitmap Join Indexes	16
KẾT LUẬN	19

ĐẶT VẤN ĐỀ

Kho dữ liệu là tuyển tập các cơ sở dữ liệu tích hợp, hướng chủ đề, được thiết kế để hỗ trợ cho chức năng trợ giúp quyết định.

Kho dữ liệu thường rất lớn tới hàng trăm GB hay thậm chí hàng Terabyte. Kho dữ liệu được xây dựng để tiện lợi cho việc truy cập theo nhiều nguồn, nhiều kiểu dữ liệu khác nhau sao cho có thể kết hợp được cả những ứng dụng của các công nghệ hiện đại và kế thừa được từ những hệ thống đã có sẵn từ trước.

Lược đồ hình sao hay lược đồ hình bông tuyết là các mô hình phổ biến của kho dữ liệu. Các truy vấn trên lược đồ loại này được gọi là truy vấn hình sao. Vì kích thước kho dữ liệu là rất lớn, chính vì vậy việc tối ưu hóa truy vấn là một vấn đề rất quan trọng nhằm nâng cao hiệu suất truy vấn, từ đó góp phần cải thiện hiệu suất của toàn bộ hệ thống.

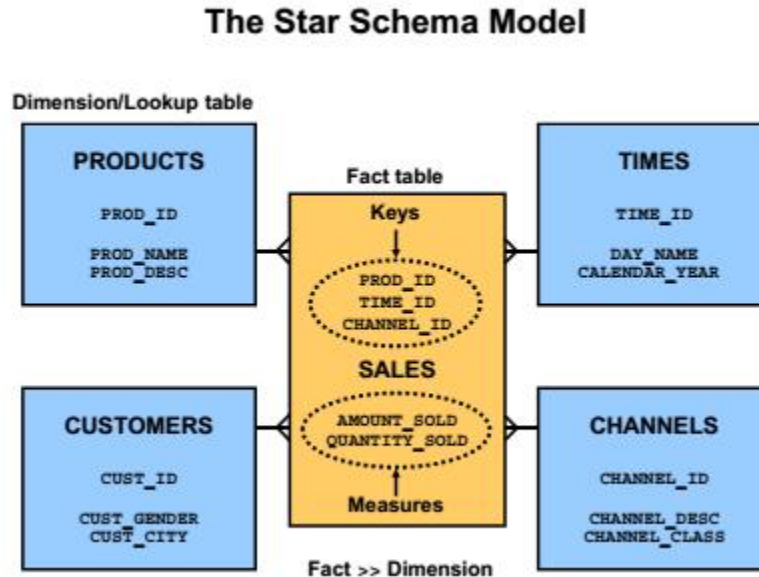
Chương 9 trong tài liệu của Oracle đã trình bày một kỹ thuật giúp cải thiện đáng kể hiệu suất của các truy vấn hình sao, đó là kỹ thuật hay thuật toán **“Star Transformation”**.

TÀI LIỆU THAM KHẢO

- [1] Chương 9 - Oracle Database 11g: SQL Tuning Workshop (Student Guide)
- [2] <http://logicalread.solarwinds.com/oracle-11g-bitmap-join-indexes-mc02/#.VUY7GlWqqkp>
- [3] https://blogs.oracle.com/optimizer/entry/star_transformation
- [4] <https://docs.oracle.com/database/121/DWHSG/schemas.htm#DWHSG9042>

I. Lược đồ trong mô hình kho dữ liệu

1. Lược đồ hình sao



Hình 1: Mô hình lược đồ hình sao

Lược đồ hình sao là lược đồ đơn giản nhất trong số các lược đồ của kho dữ liệu data warehouse.

Nó có tên lược đồ hình sao bởi vì mô hình quan hệ giữa các thực thể (các bảng) trong lược đồ này có cấu trúc hình sao.

Ở giữa hình sao này là một hoặc nhiều bảng sự kiện (fact table). Liên kết với các bảng sự kiện này là các bảng chiều (Dimension/Lookup table).

Mỗi lược đồ hình sao đặc trưng bởi 1 hoặc nhiều bảng fact chứa các thông tin chính trong kho dữ liệu và 1 tập các bảng chiều, mỗi bảng chứa thông tin về một thuộc tính cụ thể trong bảng fact.

Fact chứa các thông tin cơ sở ở mức giao tác trong nghiệp vụ mà các ứng dụng cần thiết.

Ví dụ: Trong lược đồ về lịch sử bán hàng ở trên, khi phân tích dữ liệu về sản phẩm thì cần các thông tin như: Mã sản phẩm, tên sản phẩm, mô tả về sản phẩm,... Trước khi đưa vào kho dữ liệu thì cần phải chọn mã sản phẩm và khóa ngoại trong các quan hệ liên kết và khóa ngoại này được lưu trong bảng Fact.

Dimension chứa các thông tin mô tả, các dữ liệu cần thiết cho việc thực hiện các giao tác nghiệp vụ theo một chiều hay một phạm vi nào đó.

Ví dụ: Các thông tin về sản phẩm bao gồm mã sản phẩm, tên sản phẩm, mô tả về sản phẩm, ... được lưu trong bảng Product là một bảng dimension.

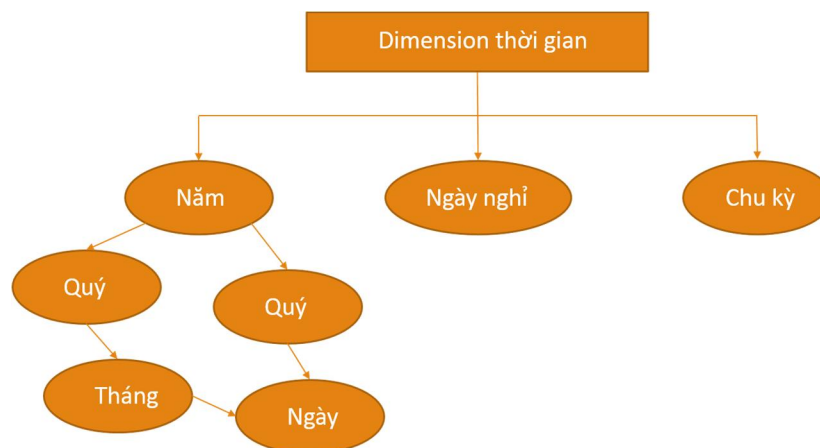
Phân loại dimension:

- Chiều phẳng (*flat dimension*): Các giá trị của chiều có cùng mức, không giá trị nào đó là cha hay là con của giá trị khác.



Hình 2: Chiều phẳng trong lược đồ hình sao

- Chiều phân cấp (*hierarchiacal dimension*): Các giá trị của chiều có quan hệ một nhiều:



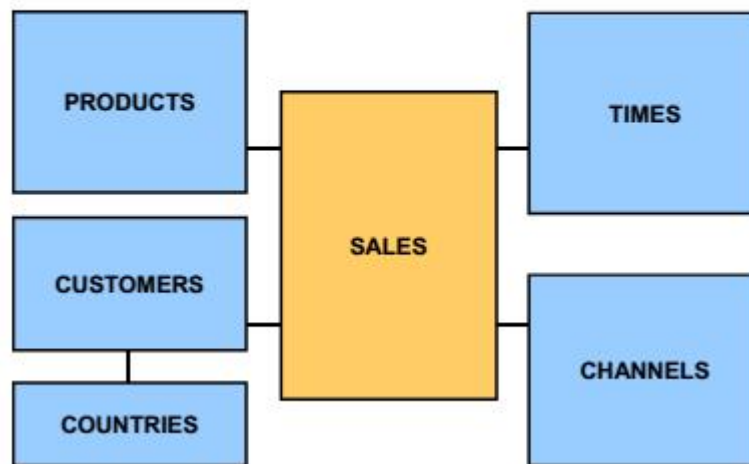
Hình 3: Chiều phân cấp trong lược đồ hình sao

Đánh giá mô hình lược đồ hình sao

- *Ưu điểm:*
 - Hỗ trợ rất đa dạng các thao tác, xử lý khá hiệu quả những câu truy vấn.
 - Phù hợp với cách mà người sử dụng nhận và sử dụng dữ liệu.
 - Nguyên lý cơ bản của lược đồ hình sao là một dạng dư thừa dữ liệu cải thiện sự thực hiện các truy vấn.
 - Trực quan, dễ sử dụng, thể hiện khung nhìn đa chiều về dữ liệu trong cơ sở dữ liệu quan hệ.
- *Nhược điểm:*
 - Việc bảng Fact được tổng hợp từ trước và được kết hợp theo chiều nên xu hướng có rất nhiều hàng và tăng trưởng một cách nhanh chóng dẫn đến độ dư thừa dữ liệu lớn.

2. Mô hình lược đồ hình bông tuyết

The Snowflake Schema Model



Hình 4: Mô hình lược đồ hình bông tuyết

Lược đồ hình bông tuyết là một lược đồ phức tạp hơn so với lược đồ hình sao, có thể coi đây là sự mở rộng của lược đồ hình sao, tại đó mỗi cánh sao không phải là một bảng dimension mà là nhiều bảng. Các bảng dimension được chuẩn hóa để loại bỏ dư thừa.

Ví dụ: Bảng products trong lược đồ hình sao có thể được chuẩn hóa và tách thành các bảng products, product_category và bảng product_manufacturer trong lược đồ hình bông tuyết. Hoặc như hình trên, bảng customers được chuẩn hóa và sử dụng bảng countries.

Đánh giá mô hình lược đồ hình bông tuyết:

- *Ưu điểm:*
 - Cải thiện năng suất truy vấn.
 - Tối thiểu không gian đĩa cần thiết để lưu trữ dữ liệu
 - Tăng tính linh hoạt của các ứng dụng bởi sự chuẩn hóa ít mang bản chất theo chiều hơn.
- *Nhược điểm:*
 - Tăng số lượng bảng dimension, cùng với đó là yêu cầu nhiều hơn số lượng khóa ngoài cho việc kết nối các bảng dimension này.

II. Star Transformation

1. Truy vấn trong lược đồ hình sao

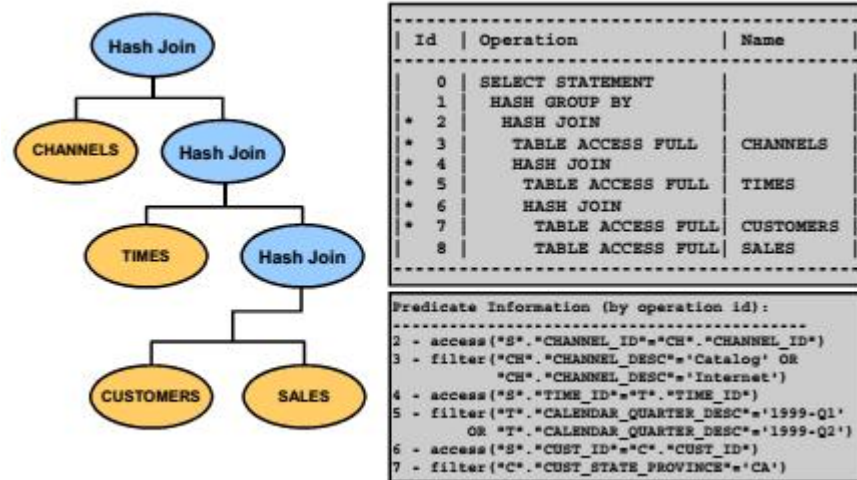
```
SELECT ch.channel_class, c.cust_city,
       t.calendar_quarter_desc,
       SUM(s.amount_sold) sales_amount
FROM sales s, times t, customers c, channels ch
WHERE s.time_id = t.time_id AND
      s.cust_id = c.cust_id AND
      s.channel_id = ch.channel_id AND
      c.cust_state_province = 'CA' AND
      ch.channel_desc IN ('Internet', 'Catalog') AND
      t.calendar_quarter_desc IN ('1999-Q1', '1999-Q2')
GROUP BY ch.channel_class, c.cust_city,
          t.calendar_quarter_desc;
```

Hình 5: Ví dụ truy vấn hình sao

Hình trên minh họa một câu truy vấn trong lược đồ hình sao với bảng sales là bảng sự kiện (Fact) và các bảng còn lại đóng vai trò là bảng chiều (Dimensions).

Bình thường để trả lời cho truy vấn trên, hệ thống sẽ sử dụng kế hoạch thực thi truyền thống, tức là kết nối tất cả các bảng lại với nhau và đưa ra danh sách các bản ghi thỏa mãn điều kiện truy vấn.

Kế hoạch thực thi không sử dụng Star Transformation:



Hình 6: Kế hoạch thực thi không sử dụng Star Transformation

Kết quả của việc kết nối này đến số lượng các bản ghi sẽ tăng lên rất nhanh dẫn đến hiệu suất truy vấn không cao, tốn tài nguyên hệ thống.

Để giải quyết vấn đề này, một kỹ thuật hay một thuật toán được đề xuất nhằm cải thiện hiệu suất của các truy vấn hình sao trong mô hình kho dữ liệu, đó là kỹ thuật “*Star Transformation*”.

2. Star Transformation là gì?

2.1. Bitmap index

Bitmap Index là một phương pháp đánh chỉ mục sử dụng cấu trúc dữ liệu đặc biệt là Bitmap. Nó được dùng để làm việc với những trường có dữ liệu rời rạc, với số lượng ít các giá trị khác nhau (tức mức độ lặp lại ở trong trường này thường là lớn), nhưng tần suất truy vấn đến trường này là rất cao.

Bitmap index được sử dụng khi:

- Mức độ trùng lặp lớn: $\text{distinct value} / \text{total value} < 1\%$. Ví dụ: trường giới tính $\text{distinct} = 2$, giả sử $\text{total value} = 1000 \rightarrow 2/1000 < 1\%$.

- Không có hoặc ít thao tác update hoặc insert lên bảng dữ liệu: Vì cách tổ chức của bitmap phức tạp nên việc update lại thông tin của Bitmap index phải trả 1 chi phí lớn cho việc duy trì cấu trúc của nó. Nó chiếm nhiều thời gian, không gian ổ cứng và thiết bị nhập xuất. Người thiết kế phải đảm bảo rằng giữ không để vượt quá phạm vi kích thước của chỉ mục.

- Khi có nhiều cột tham gia index: Một trong những lợi thế lớn nhất của bitmap index là có thể có nhiều cột trong đó nên nó có thể kết hợp các cột dữ liệu lại với nhau.

2.2. Định nghĩa Star Transformation

Ý tưởng cơ bản của “Star Transformation” là chỉ truy cập tới danh sách các hàng cần thiết trong bảng fact table thay vì việc phải quét qua toàn bộ bảng này để trả lời một truy vấn hình sao.

Truy vấn hình sao là thuật ngữ dùng để chỉ chung các truy vấn trên lược đồ hình sao, lược đồ hình bông tuyết.

Để thực hiện ý tưởng trên, các điều kiện được đặt ra cho kho dữ liệu gồm:

- Xây dựng một bitmap indexes trên mỗi trường khóa ngoại thuộc bảng sự kiện (fact table).
- Tham số STAR_TRANSFORMATION_ENABLED được khởi tạo giá trị TRUE.

Một khi kho dữ liệu thỏa mãn hai điều kiện trên, đa số các truy vấn hình sao sẽ được thực hiện với một chiến lược thực thi được biết đến với tên gọi “Star Transformation”. Star Transformation giúp cải thiện hiệu suất của các truy vấn hình sao một cách hiệu quả.

Star Transformation là một kỹ thuật tối ưu hóa mạnh mẽ dựa trên việc viết lại (hay chuyển đổi) các câu truy vấn SQL từ câu truy vấn gốc. Việc chuyển đổi này hoàn toàn trong suốt đối với người sử dụng, nó được thực hiện một cách hoàn toàn tự động bởi hệ thống khi thích hợp.

Một kế hoạch thực thi cho một truy vấn hình sao sử dụng Star Transformation diễn ra trong hai pha:

- Pha thứ nhất, thu thập các bản ghi cần thiết từ bảng sự kiện (fact table). Việc thu thập này sử dụng bitmap index, nên có hiệu suất cao hơn.
- Pha thứ hai, kết nối tập kết quả từ pha thứ nhất với các bảng dimension. Hành động này có tên gọi semi-join.

Star Transformation yêu cầu ít nhất 2 bảng dimensions và một bảng sự kiện.

2.3. Lưu ý

Star Transformation không hỗ trợ cho các trường hợp sau:

- Truy vấn tới bảng không tương thích với việc truy cập thông qua bitmap
- Các truy vấn có chứa biến
- Bảng sự kiện bên ngoài
- Chứa bảng không cho phép kết nối
- ...

3. Quy trình thực hiện Star Transformation

3.1. Viết lại câu truy vấn

Phase 1

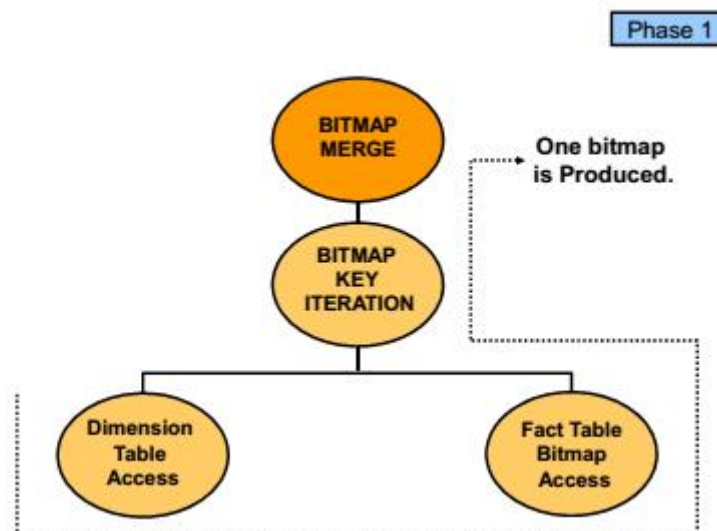
```

SELECT s.amount_sold
FROM sales s
WHERE time_id IN (SELECT time_id
                   FROM times
                   WHERE calendar_quarter_desc
                       IN('1999-Q1', '1999-Q2'))
AND cust_id IN (SELECT cust_id
                 FROM customers
                 WHERE cust_state_province = 'CA')
AND channel_id IN (SELECT channel_id
                   FROM channels
                   WHERE channel_desc IN
                       ('Internet', 'Catalog'));
    
```

Hình 7: Câu truy vấn được viết lại

Đối với một truy vấn hình sao như đã trình bày trong hình (). Hệ thống sẽ viết lại câu truy vấn như trên, sử dụng một bộ lọc trên bảng dimension để thu thập danh sách khóa trên bảng này thỏa mãn bộ lọc. Sau đó, tạo ra một bitmap index trên trường khóa ngoại tương ứng trong bảng Fact để định danh và thu thập những bản ghi phù hợp.

3.2. Pha thứ nhất – thu thập các bản ghi từ một bảng dimension



Hình 8: Thu thập các bản ghi từ một bảng Dimension

Đầu tiên, hệ thống sẽ truy cập vào bảng dimension, lọc ra danh sách các bản ghi thỏa mãn bộ lọc. Sau đó với mỗi bản ghi phù hợp, hệ thống tìm kiếm chỉ mục bitmap tương ứng với giá trị key của bản ghi đó trong bảng bitmap index tương ứng với chiều đang xét và lấy nó về.

“BITMAP KEY ITERATION” là một vòng lặp quá trình trên với tất cả các giá trị thỏa mãn bộ lọc. Kết quả thu được sau vòng lặp này là một tập hợp các bitmap index của các giá trị từ bảng dimension thỏa mãn bộ lọc ban đầu.

Bước cuối cùng là kết hợp tất cả các bitmap index thu được sau vòng lặp để thu được một bitmap duy nhất đại diện cho danh sách bản ghi thuộc bảng Fact chứa khóa ngoại liên kết với những bản ghi thỏa mãn bộ lọc.

Như vậy đến đây, ta đã hoàn thành việc thu thập các bản ghi thỏa mãn từ một bảng Dimension.

Ví dụ: Xét điều kiện sau:

```
WHERE time_id IN (SELECT time_id
                  FROM times
                  WHERE calendar_quarter_desc
                     IN ('1999-Q1', '1999-Q2'))
```

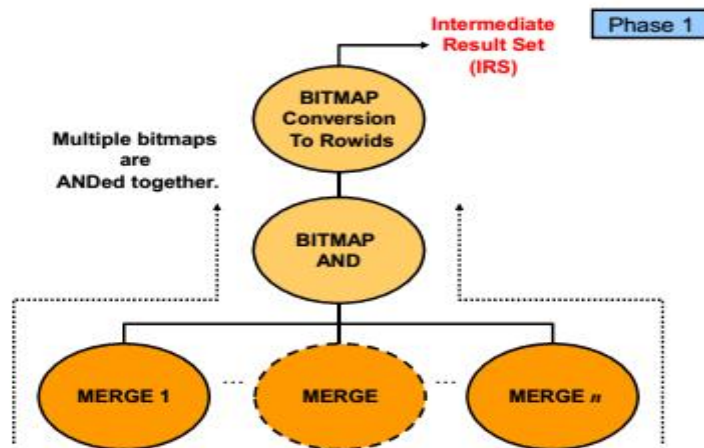
Ta sẽ xét chiều time, giá trị calendar_quarter_desc thuộc một trong 2 giá trị ‘1999-Q1’ hoặc ‘1999-Q2’ tương ứng với 2 giá trị khóa pk1 và pk2.

Giả sử khóa ngoại trong bảng Fact tương ứng với 2 giá trị khóa trên là fk1 và fk2. Bảng bitmap index cho chiều time đối với bảng Fact sẽ có dạng:

...	...
fk1	[1, 0, 0, 0, ...]
fk2	[0, 0, 1, 0, ...]
...	...

Sau bước thứ nhất trên, ta sẽ thu được 1 bitmap duy nhất là [1, 0, 1, 0, ...] đại diện cho các bản ghi thỏa mãn.

3.3. Pha thứ nhất – thu thập các bản ghi từ tất cả bảng dimension

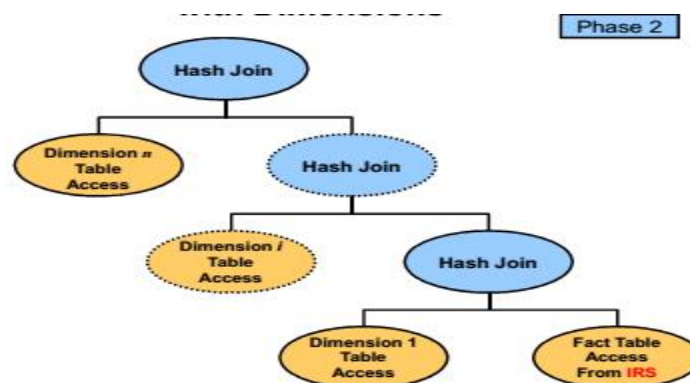


Hình 9: Thu thập các bản ghi từ tất cả bảng dimension

Quá trình được trình bày ở trên sẽ được lặp lại với tất cả các chiều. Mỗi bước BITMAP MERGE sẽ tạo ra một bitmap tương ứng với một chiều như đã trình bày. Tiếp theo hệ thống sẽ phải thực hiện việc kết nối tất cả kết quả đó lại để thu được một bitmap duy nhất đại diện cho các bản ghi thỏa mãn tất cả các bộ lọc tương ứng với mỗi chiều. Để thực hiện việc này, hệ thống sử dụng phép toán AND BITMAP.

Cho đến thời điểm hiện tại, hệ thống chỉ mới truy cập tới các bảng dimension và bảng chứa bitmap index của bảng Fact. Để có thể truy cập vào bảng Fact, cần thực hiện việc chuyển đổi từ bitmap thu được ở trên (tức là kết quả của phép BITMAP AND) thành chỉ số hàng, kết quả thu được là tập kết quả trung gian (Intermediate Result Set – IRS)

3.4. Pha thứ hai - Kết nối tập kết quả trung gian với các bảng dimensions



Hình 10: Kết nối tập kết quả trung gian với các bảng Dimension

Sau khi có được tập kết quả trung gian IRS, hệ thống chuyển sang pha thứ hai trong thuật toán “Star Transformation”.

Trong pha này , cần thực hiện kết nối dữ liệu từ bảng Fact, tương ứng là tập kết quả, với các dữ liệu trong bảng Dimension.

Trong sơ đồ trên, để kết nối bảng Fact với các bảng Dimension, ta sử dụng Hash join, đây là phương pháp kết nối phổ biến nhất để kết nối các hàng trong một truy vấn hình sao, tuy nhiên nó không phải luôn luôn chính xác.

Ta có kế hoạch thực thi của một truy vấn hình sao khi áp dụng Star Transformation như sau:

```

SORT GROUP BY
HASH JOIN
  HASH JOIN
    TABLE ACCESS BY INDEX ROWID SALES
    BITMAP CONVERSION TO ROWIDS
    BITMAP AND
    BITMAP MERGE
    BITMAP KEY ITERATION
    BUFFER SORT
    TABLE ACCESS FULL CHANNELS
    BITMAP INDEX RANGE SCAN SALES_CHANNELS_BX
    BITMAP MERGE
    BITMAP KEY ITERATION
    BUFFER SORT
    TABLE ACCESS FULL TIMES
    BITMAP INDEX RANGE SCAN SALES_TIMES_BX
    ...
    TABLE ACCESS FULL CHANNELS
    TABLE ACCESS FULL TIMES
  
```

Hình 11: Kế hoạch thực thi truy vấn hình sao kết hợp Star Transformation

4. Cải thiện hiệu suất cho “Star Transformation”

4.1. Sử dụng bảng trung gian

Khi nhìn vào kế hoạch thực thi trong Star Transformation, ta thấy rằng mỗi bảng Dimension đều được truy cập tới hai lần, lần thứ nhất trong pha một giúp tìm ra tập các bản ghi thỏa mãn điều kiện truy vấn trong bảng Fact, lần thứ hai kết nối tập kết quả trung gian từ pha thứ nhất với các bảng Dimension để đưa ra trả lời cho câu truy vấn.

Điều này làm giảm hiệu suất truy vấn trong trường hợp các bảng Dimension có kích thước lớn và không có phương pháp truy cập tối ưu tới các bảng này.

Để giải quyết vấn đề này, hệ thống sẽ quyết định tạo ra một bảng tạm chứa dữ liệu cần thiết cho cả hai pha trong trường hợp chi phí tạo ra bảng tạm này là nhỏ hơn so với việc quét hai lần qua toàn bộ các bản ghi trong bảng Dimension.

Ví dụ: Kế hoạch sinh bảng trung gian của hệ thống:

```
LOAD AS SELECT      SYS_TEMP_0FD9D6720_BEBDC
TABLE ACCESS FULL   CUSTOMERS
...
|
filter("C"."CUST_STATE_PROVINCE"='CA')
```

Lưu ý: Hệ thống sẽ không tự quyết định sử dụng các bảng trung gian trong các trường hợp sau:

- Cơ sở dữ liệu ở dạng chỉ đọc
- Truy vấn hình sao là một phần của một giao dịch trong chế độ serializable.
- Tham số STAR_TRANSFORMATION_ENABLED được thiết lập với giá trị TEMP_DISABLE.

4.2. Sử dụng Bitmap Join Indexes

Thay vì việc sử dụng bitmap index trên một bảng duy nhất, ta có thể xây dựng một bitmap index trên 2 hay nhiều bảng được kết nối với nhau, được gọi là “bitmap join index”. Chỉ mục bitmap trên bảng Fact được xây dựng dựa trên dữ liệu tương ứng trên bảng Dimension.

Khi đó thay vì phải quét qua toàn bộ bảng Dimension trong pha thứ nhất, hệ thống chỉ cần truy cập vào bitmap join index được xây dựng.

Ví dụ:

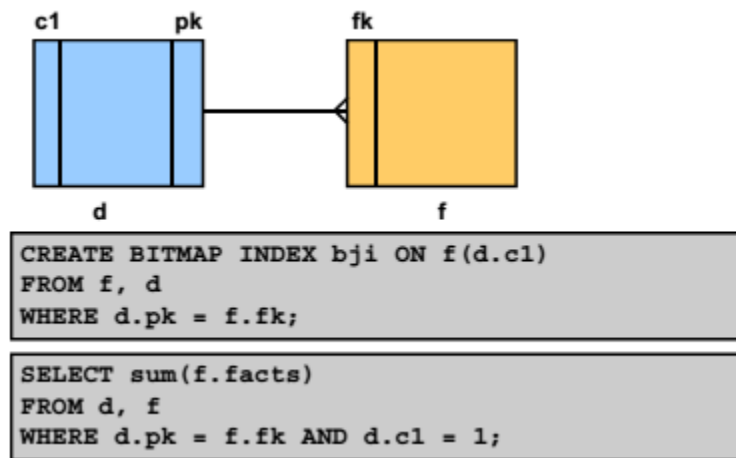
```
CREATE BITMAP INDEX sales_q_bjx
ON sales(times.calendar_quarter_desc)
FROM sales, times
WHERE sales.time_id = times.time_id LOCAL;
```

```
SORT GROUP BY
HASH JOIN
  HASH JOIN
    TABLE ACCESS BY INDEX ROWID SALES
    BITMAP CONVERSION TO ROWIDS
    BITMAP AND
      BITMAP MERGE
        BITMAP KEY ITERATION
        BUFFER SORT
        TABLE ACCESS FULL CHANNELS
        BITMAP INDEX RANGE SCAN SALES_CHANNELS_BX
      BITMAP OR
        BITMAP INDEX SINGLE VALUE SALES_Q_BJX
        BITMAP INDEX SINGLE VALUE SALES_Q_BJX
    TABLE ACCESS FULL CHANNELS
  TABLE ACCESS FULL TIMES
```

Hình 12: Kế hoạch thực thi truy vấn khi áp dụng bitmap join index

4.3. Các mô hình kết nối của Bitmap Join Indexes

Mô hình 1: Một cột trên bảng Dimension kết nối với bảng Fact

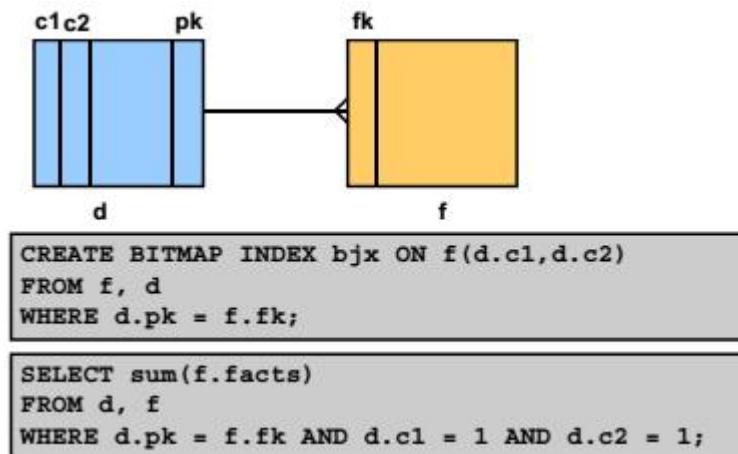


Hình 13: Một cột trên bảng Dimension kết nối với bảng Fact

Bitmap join index được sử dụng trong mệnh đề SELECT để tránh phép toán kết nối giữa 2 bảng

Hệ thống sẽ tính toán trước kết quả kết nối giữa fk với c1 và lưu vào một bitmap index thay vì việc kết nối hai bảng và lưu vào một bảng thứ ba.

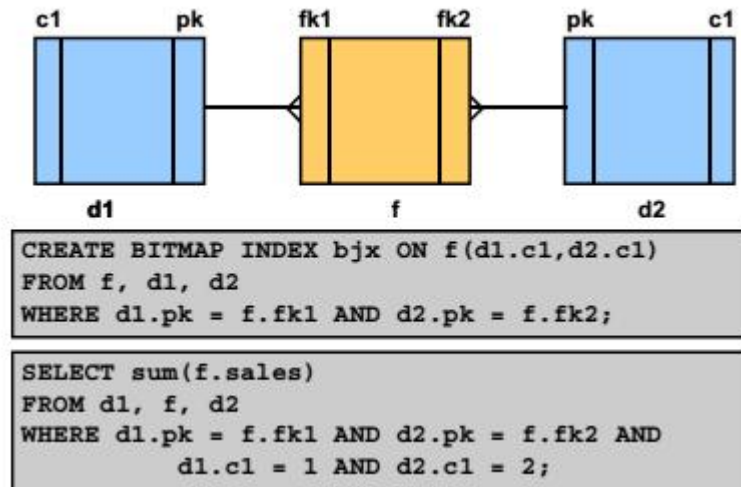
Mô hình 2: Nhiều cột trên bảng Dimension kết nối với bảng Fact



Hình 14: Nhiều cột trên bảng Dimension kết nối với bảng Fact

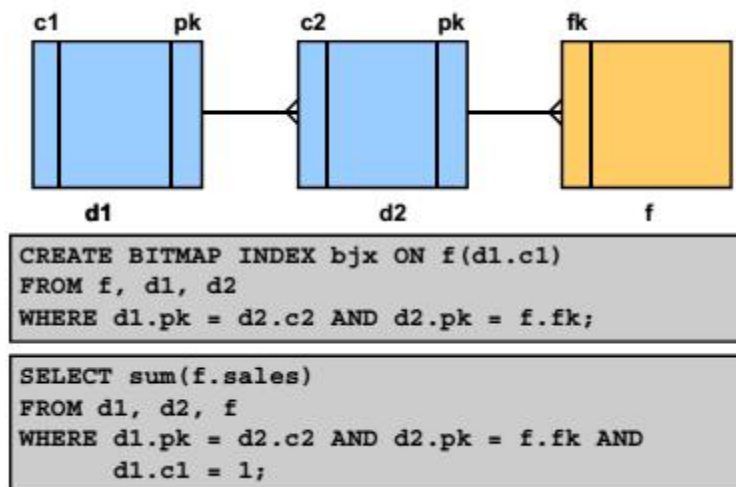
Đây là mô hình mở rộng so với mô hình thứ nhất, cho phép tạo bitmap join index trên nhiều cột của bảng Dimension cùng với bảng Fact.

Mô hình 3: Nhiều bảng Dimension kết nối với bảng Fact



Hình 15: Một bảng Dimension kết nối với bảng Fact

Mô hình 4: Bitmap join index trong lược đồ hình bông tuyết



Hình 16: Bitmap join indexes trên lược đồ hình bông tuyết

Những hạn chế của Bitmap Join Indexes:

- Chỉ một bảng được cập nhật đồng thời bởi các giao dịch khác nhau khi sử dụng bitmap join index.
- Một bảng không thể xuất hiện hai lần khi kết nối.
- Không thể tạo bitmap join index trên bảng tạm.
- Tất cả các cột trong chỉ mục đều phải là cột trong bảng dimension.
- Cột dùng để kết nối từ bảng Dimension phải là cột chứa khóa chính hoặc ràng buộc duy nhất (unique constraints).
- ...

KẾT LUẬN

Chương 9 trong tài liệu này đã trình bày một kỹ thuật giúp nâng cao hiệu suất truy vấn đối với lược đồ hình sao cũng như lược đồ hình bông tuyết trong mô hình kho dữ liệu.

Với việc áp dụng thuật toán “Star Transformation” kết hợp với các kỹ thuật cải thiện hiệu suất khác như sử dụng bảng tạm, sử dụng Bitmap Join Index sẽ giúp cải thiện đáng kể hiệu suất truy vấn dữ liệu, góp phần nâng cao hiệu suất của toàn hệ thống.