



# data warehouse

GROUP 8



# **NỘI DUNG**

**01** STEPS TO CREATE DIMENSIONAL MODELING

**02** OLAP OPERATIONS

**03** SLOWLY CHANGING DIMENSIONS

**04** RAPIDLY CHANGING DIMENSIONS



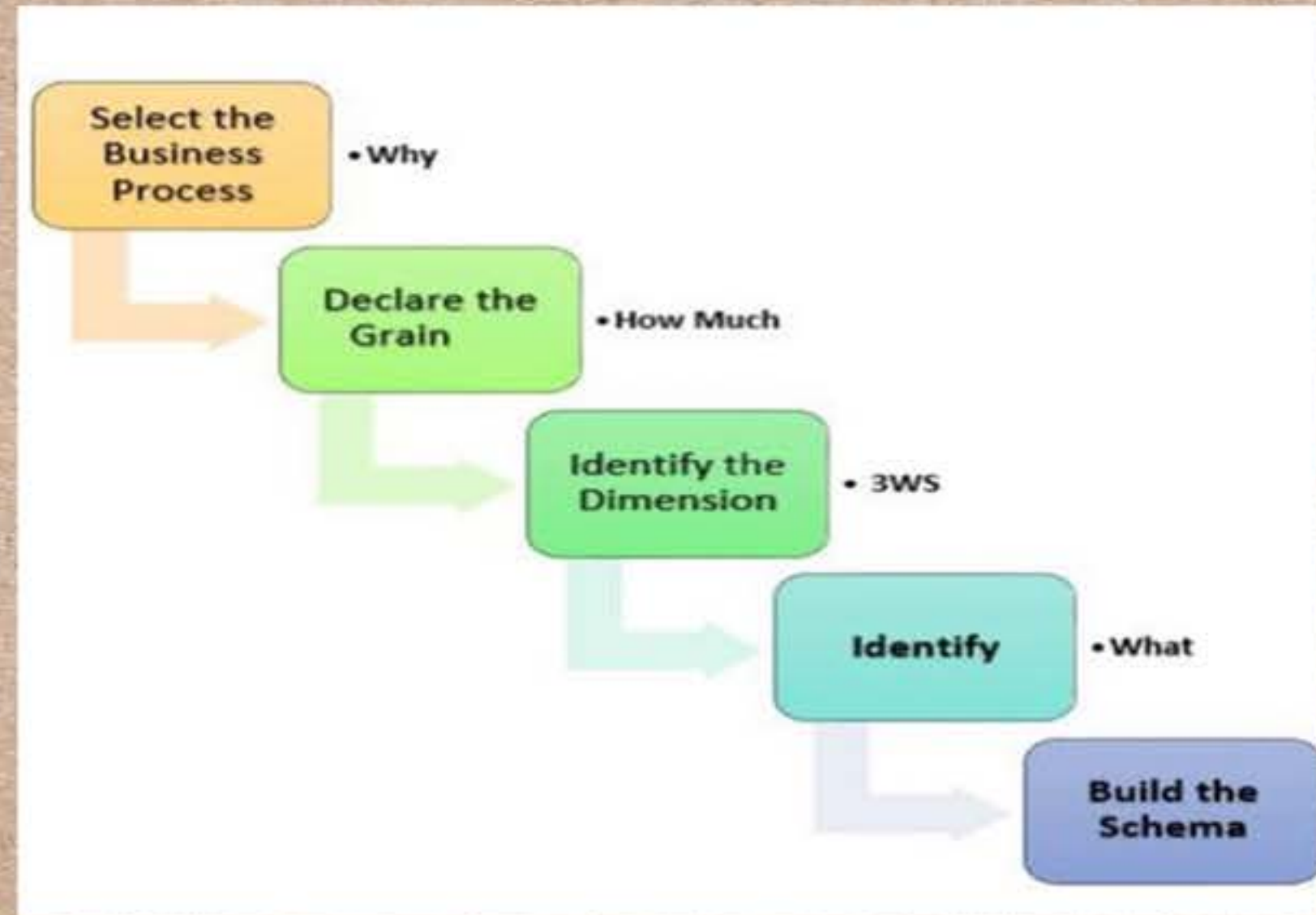
# **PHẦN 1: STEPS TO CREATE DIMENSIONAL MODELING**



# STEPS TO CREATE DIMENSIONAL MODELING

Độ chính xác trong việc tạo mô hình chiều quyết định sự thành công của việc triển khai kho dữ liệu

1. IDENTIFY BUSINESS PROCESS
2. IDENTIFY GRAIN (LEVEL OF DETAIL)
3. IDENTIFY DIMENSIONS
4. IDENTIFY FACTS
5. BUILD SCHEMA





# 1. IDENTIFY BUSINESS PROCESS

**Đây là bước quan trọng nhất của mô hình hóa dữ liệu**

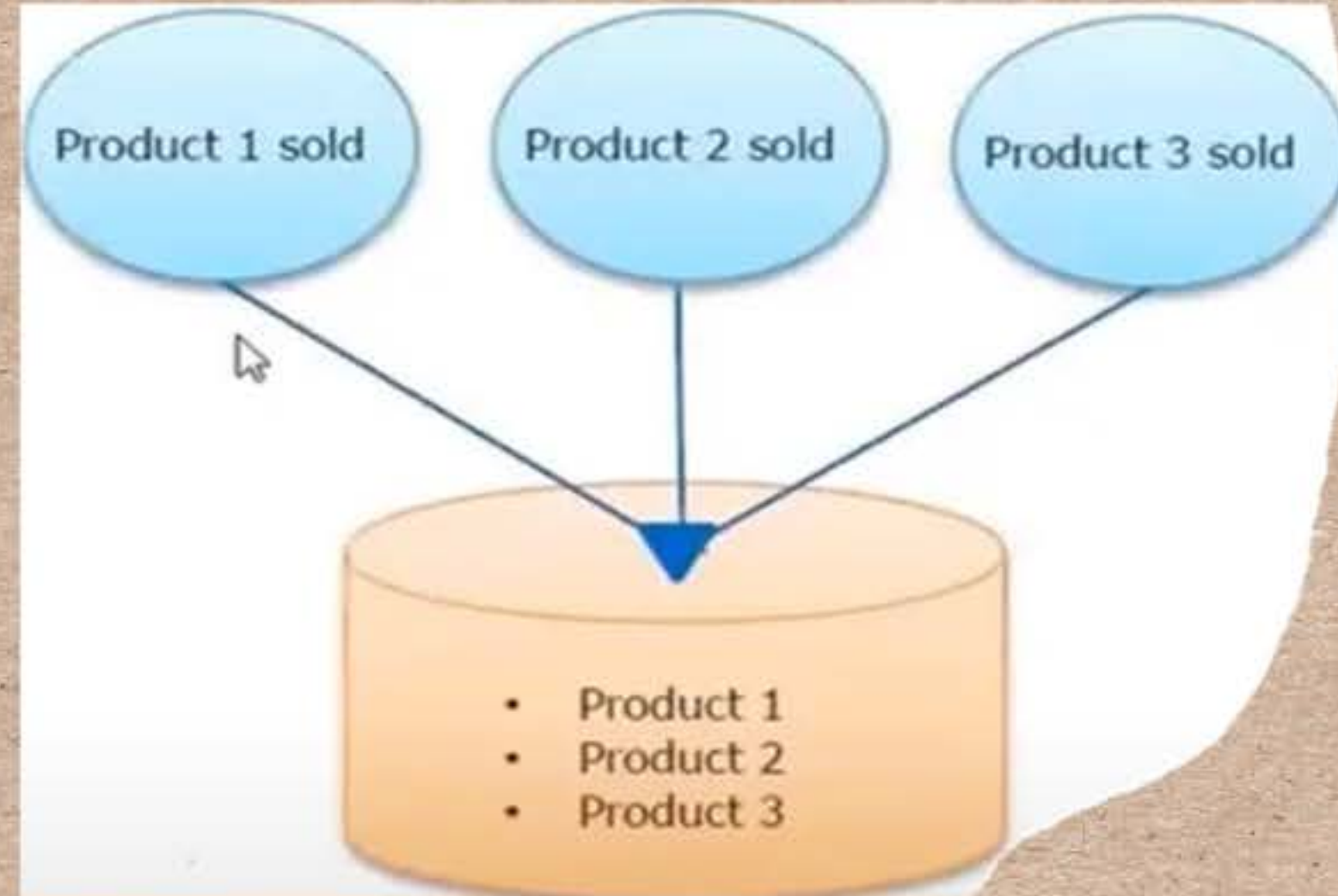
**Có 3 loại Business Process ứng với 3 loại bảng Fact**

- Transaction (giao dịch)
- Accumulating Snapshot (ảnh chụp tích lũy)
- Periodic Snapshot (ảnh chụp định kỳ)



# TRANSACTION:

- **Cơ bản nhất**
- Mỗi dòng trong bảng fact ứng với 1 transaction.
- Giống tính chất của DW, khi thông tin được đưa vào rồi sẽ **không sửa, xóa được**.
- Thông tin sẽ đưa vào bảng Fact khi transaction xảy ra.





# ACCUMULATING SNAPSHOT

- Ít được sử dụng nhưng trong một số trường hợp thì cần dùng
- Được sử dụng để nắm quy trình làm việc của Business Process
- Các hàng sẽ được chèn vào và update khi một mốc thời điểm nào đó xảy ra.

Accumulated Fact Table

Step 1	Order date		
Step 2	Order date	Shipment date	
Step 3	Order date	Shipment date	Delivered date



# PERIODIC SNAPSHOT

- **Lưu lại thông tin định kỳ** được xác định trước theo từng ngày, từng tuần, từng tháng,...
- Tại các khoảng thời gian xác định trước, các ảnh chụp có cùng mức độ chi tiết sẽ được chụp và xếp chồng lên bảng Fact.



## 2. IDENTIFY GRAIN

- Mô tả mức độ chi tiết của vấn đề/giải pháp kinh doanh.
- Cung cấp mức thông tin thấp nhất cho bất kỳ bảng nào trong Data Warehouse.

**Chúng ta có cần lưu trữ tất cả các sản phẩm có sẵn hay chỉ một vài loại sản phẩm?**

**Chúng ta có lưu trữ thông tin bán sản phẩm hàng tháng, hàng tuần, hàng ngày hay hàng giờ không?**

**Hai lựa chọn trên ảnh hưởng đến kích thước cơ sở dữ liệu như thế nào?**



### 3. IDENTIFY DIMENSIONS

- Các chiều sẽ cung cấp bối cảnh cho các dữ kiện.
- Có thể dễ dàng xác định các chiều nhờ các từ “by” và/hoặc “for”. (ngày tháng, kho, hàng tồn kho, ...)

**VD: TÌM DOANH SỐ HÀNG NGÀY CHO CÁC SẢN PHẨM  
CỤ THỂ Ở NHỮNG NƠI KHÁC NHAU**

- |                          |  |
|--------------------------|--|
| • <b>CHIỀU:</b>          | <b>Sản phẩm, Thời gian, Địa điểm</b>               |
| • <b>CÁC THUỘC TÍNH:</b> | <b>Sản phẩm:</b> Mã số, Tên, Loại                  |
| • <b>PHÂN CẤP:</b>       | <b>Địa điểm:</b> Quốc gia, Tỉnh/Thành phố, Địa chỉ |



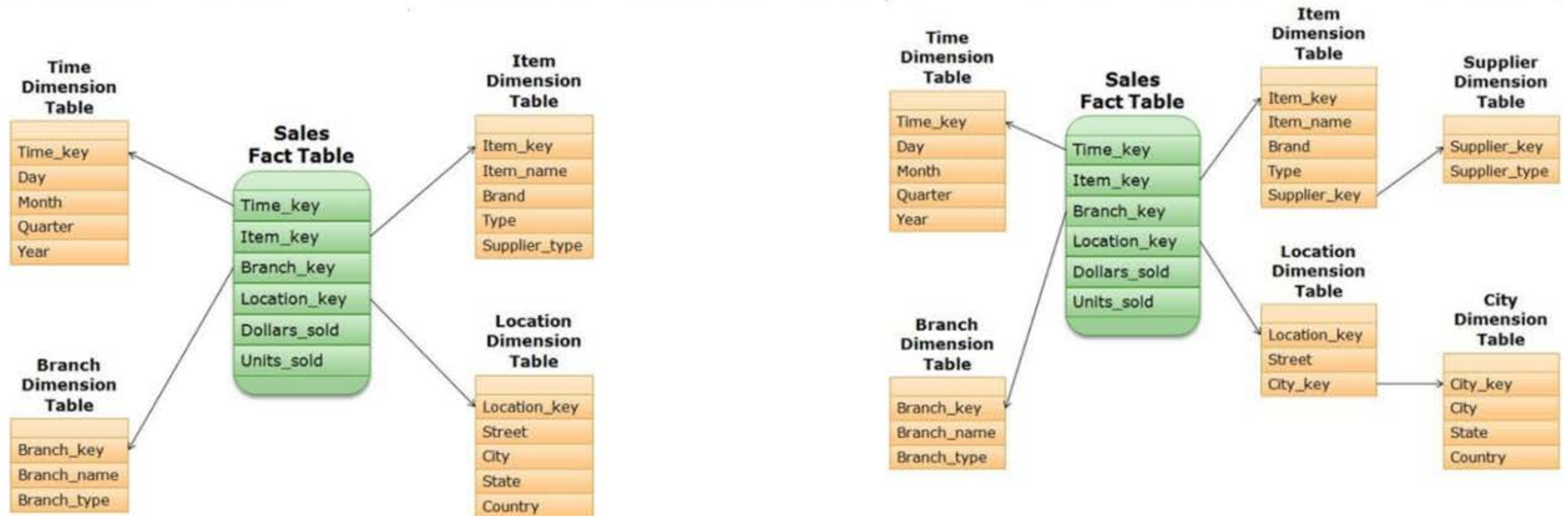
## 4. IDENTIFY FACTS

- Fact là những giá trị số có thể định lượng được gắn liền với quy trình kinh doanh.
- Được liên kết với người dùng doanh nghiệp (có quyền truy cập vào dữ liệu).
- Hầu hết các hàng của bảng dữ kiện là các giá trị số (giá) hoặc chi phí trên một đơn vị,...



# 5. BUILD SCHEMA

## Tùy chọn giữa Star Schema và Snowflake Schema





# NGUYÊN TẮC

- Nạp dữ liệu atomic (nguyên tử) vào các cấu trúc chiều.
- Xây dựng các mô hình chiều xoay quanh quy trình kinh doanh.
- Mỗi bảng fact đều có một bảng chiều date liên quan.
- Tất cả các dữ kiện trong một bảng fact đều có cùng mức độ chi tiết.
- Bảng chiều sử dụng khóa thay thế (không liên hệ với dữ liệu).
- Cân bằng giữa yêu cầu và thực tế để đưa ra giải pháp kinh doanh nhằm hỗ trợ ra quyết định.



# **PHẦN 2: OLAP OPERATIONS**



# OLAP OPERATIONS

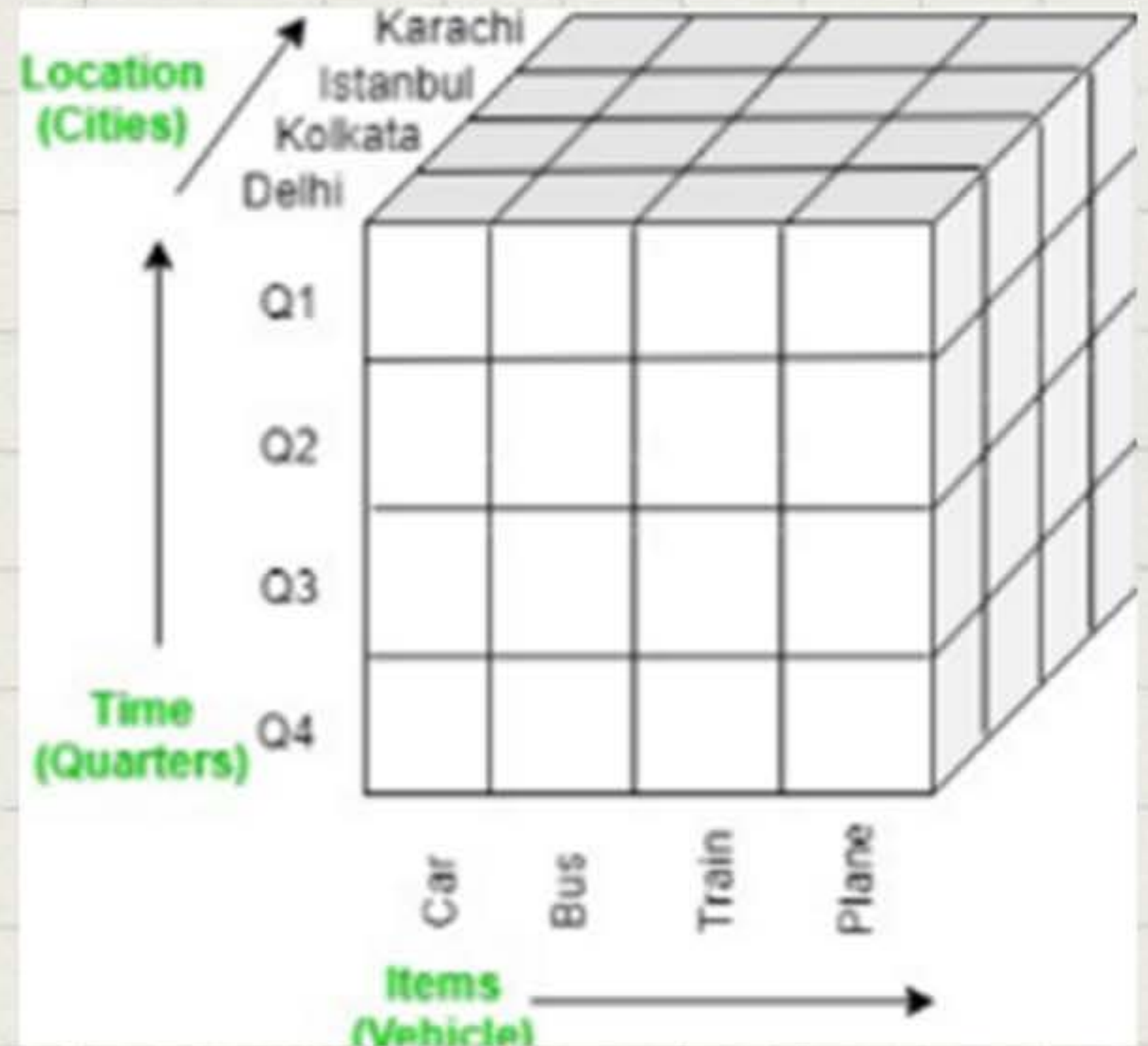
OLAP OPERATIONS (Hoạt động xử lý phân tích trực tuyến) đề cập đến hành động thực hiện các **phép toán phân tích** trên hệ thống OLAP. Các phép toán phân tích này nhằm **hỗ trợ ra quyết định** và **phân tích dữ liệu lịch sử**.





# ROLL - UP

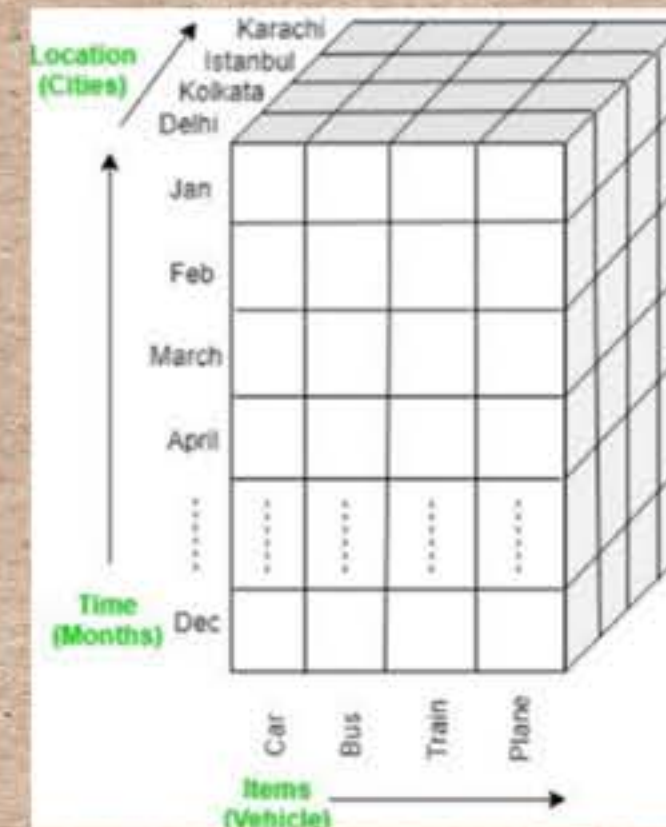
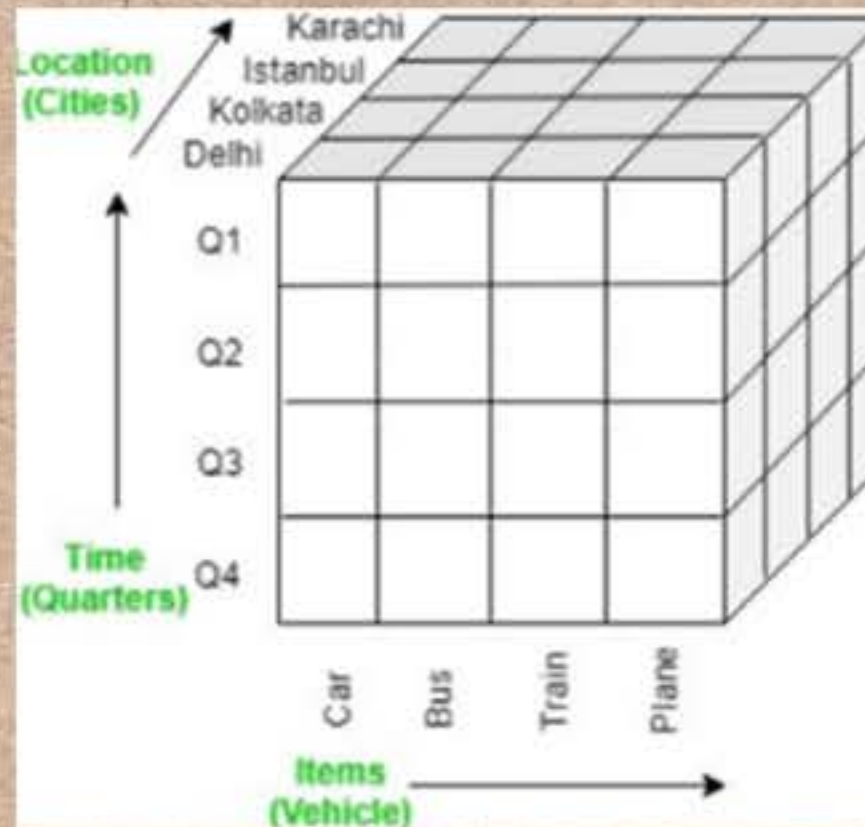
Là một kỹ thuật **giảm kích thước** trên một khối dữ liệu nhất định. Giảm kích thước có thể được thực hiện bằng cách **kết hợp kích thước tương tự** trên bất kỳ trục nào của khối dữ liệu bằng cách sử dụng khái niệm phân cấp khái niệm.





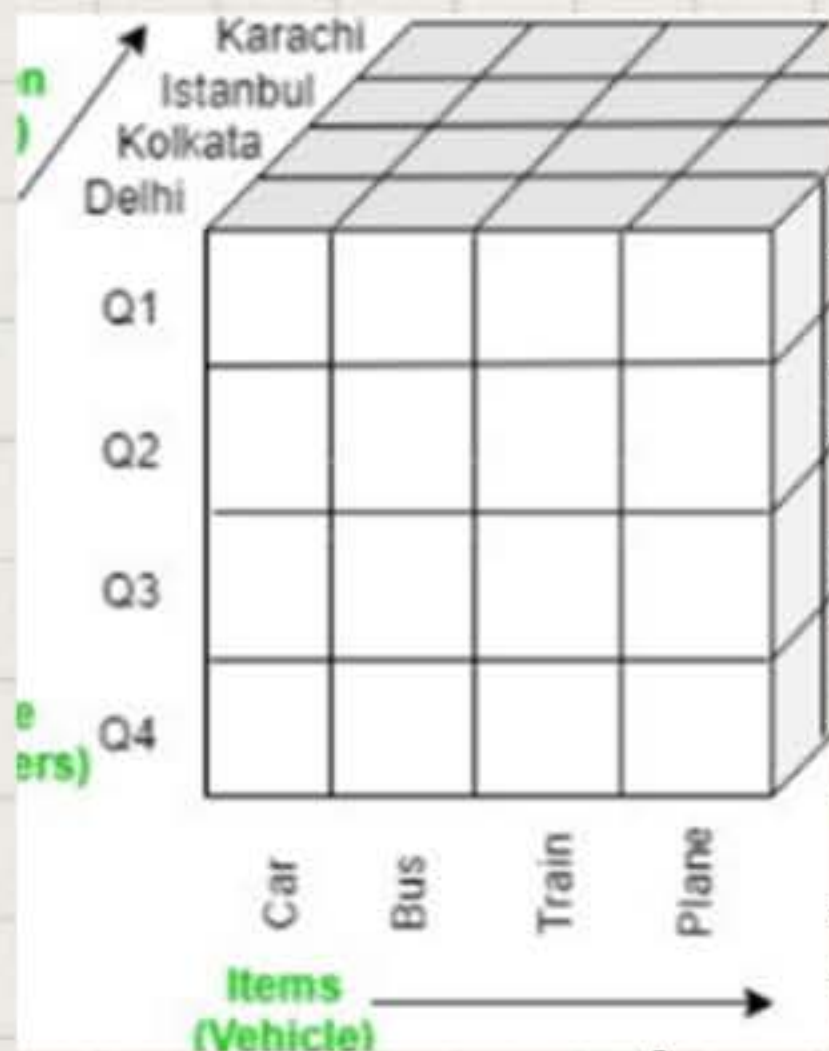
# DRILL - DOWN

**Truy sâu xuống** là một kỹ thuật mở rộng kích thước có thể được áp dụng trên khối dữ liệu. **Mở rộng kích thước** có nghĩa là, thêm thứ nguyên mới hoặc mở rộng kích thước hiện có trên bất kỳ trục nào của khối dữ liệu bằng cách sử dụng khái niệm phân cấp khái niệm.





# SLICE



LIỆT  
HẮT  
PHẦN  
DẠP

C TẠO

RA KHỐI LẬP  
PHƯƠNG PHỤ MỞI.

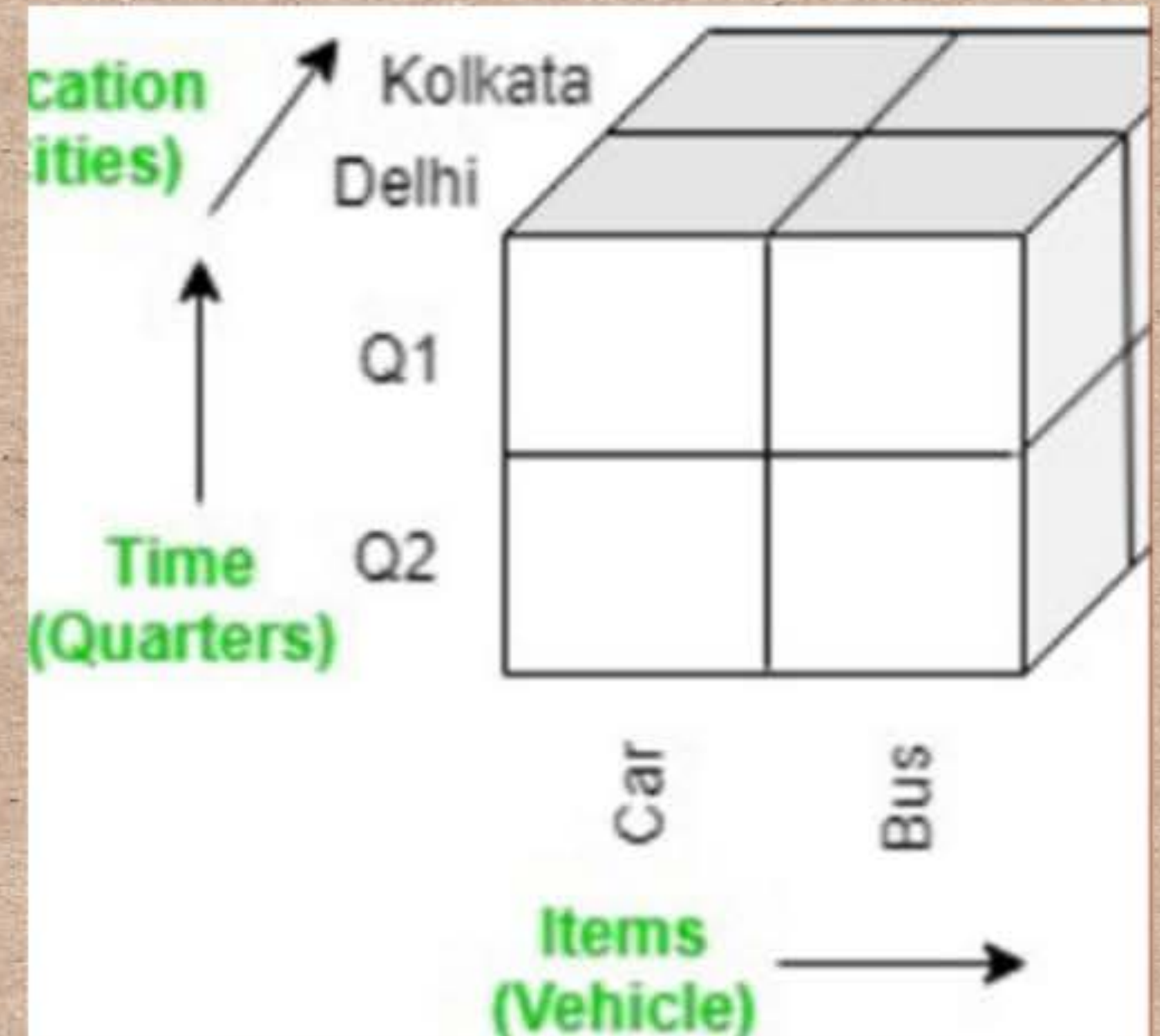
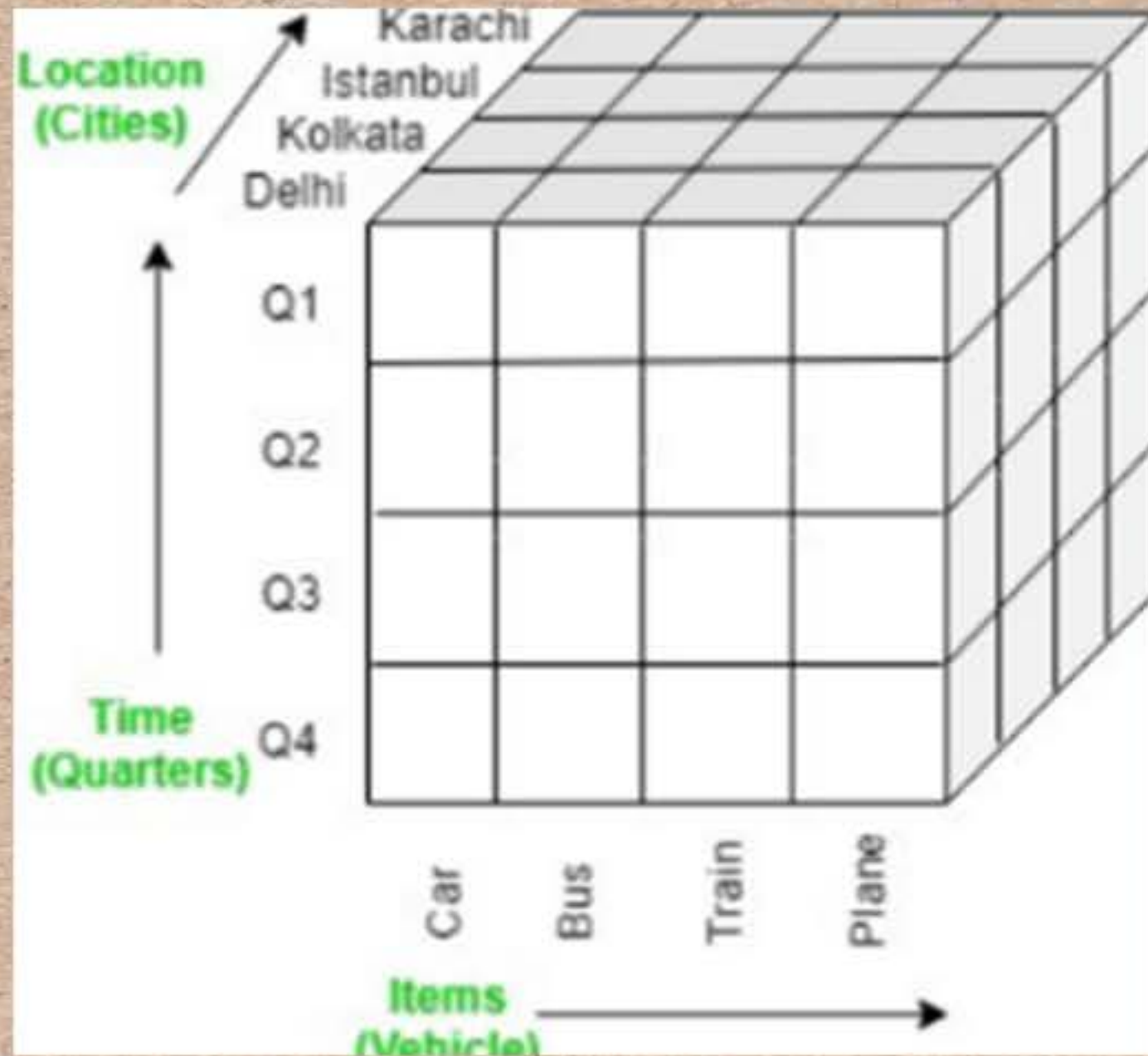
Karachi			
Istanbul			
Kolkata			
Delhi			
	Car	Bus	Train

Items (Vehicle)



# DICE

Thông qua hoạt động Dice, một khối phụ có thể được tạo ra bằng cách **chọn hai hoặc nhiều hơn hai chiều** từ khối dữ liệu.





# PIVOT

- Trong khối phụ thu được sau Slice, thực hiện thao tác Pivot cho một cái nhìn mới về nó.
- Có thể chứa việc hoán đổi các hàng và cột hoặc di chuyển một trong các kích thước hàng vào kích thước cột.

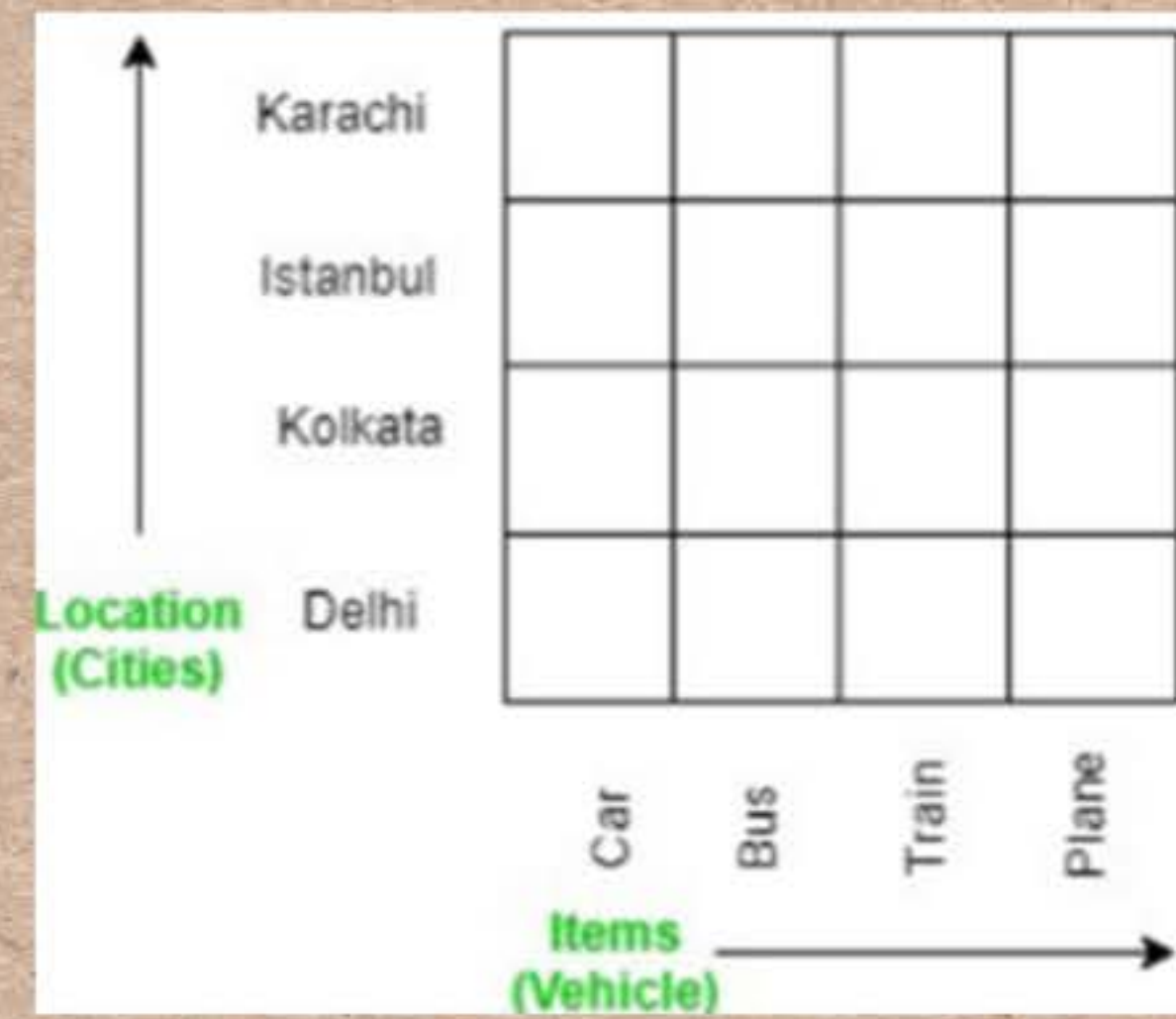


Diagram illustrating a Pivot table structure. The vertical axis is labeled "Location (Cities)" and lists four cities: Karachi, Istanbul, Kolkata, and Delhi. The horizontal axis is labeled "Items (Vehicle)" and lists four vehicle types: Car, Bus, Train, and Plane. The table is a 4x4 grid of empty cells.

Location (Cities)	Car	Bus	Train	Plane
Karachi				
Istanbul				
Kolkata				
Delhi				

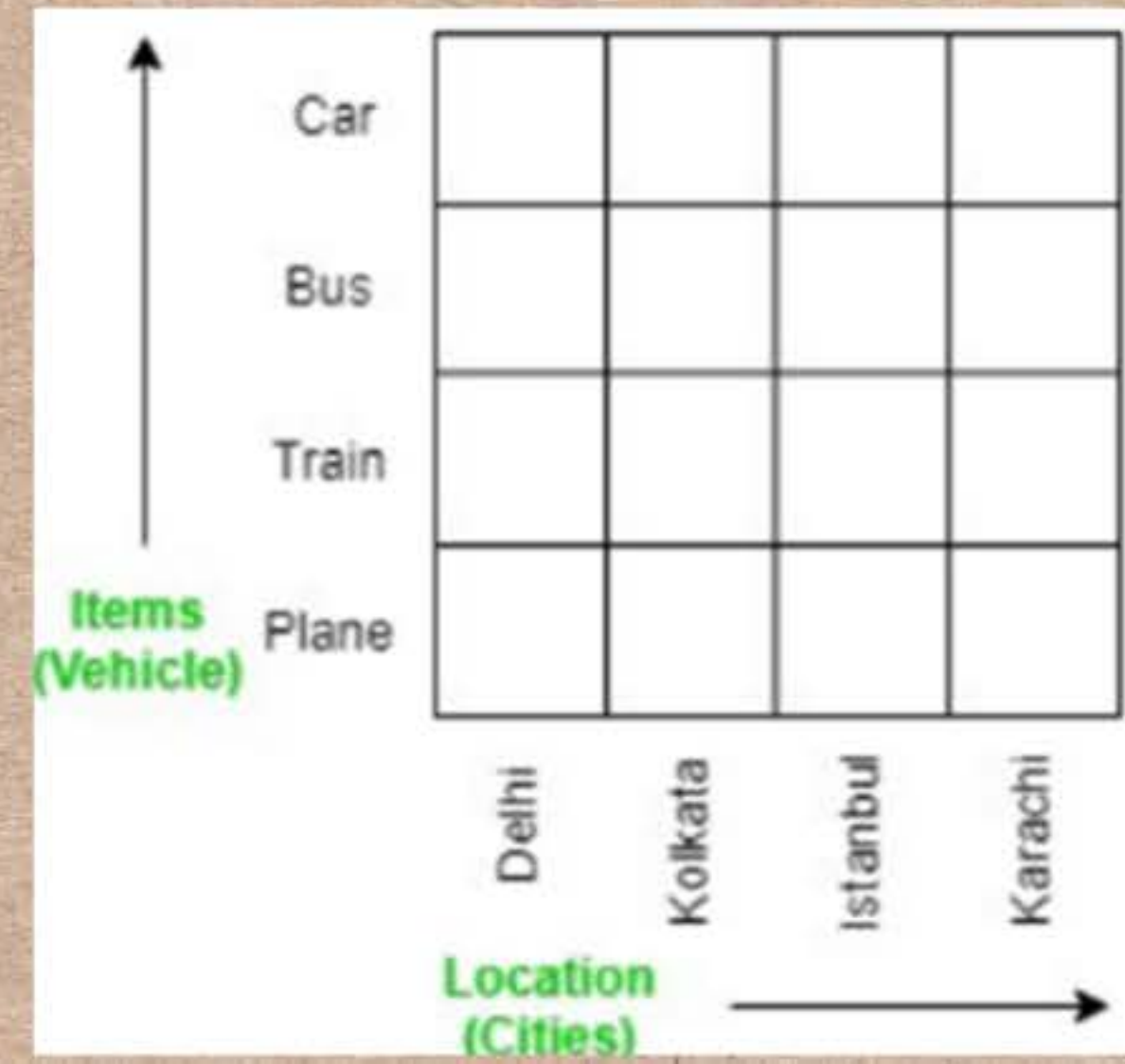


Diagram illustrating a Pivot table structure. The vertical axis is labeled "Items (Vehicle)" and lists four vehicle types: Car, Bus, Train, and Plane. The horizontal axis is labeled "Location (Cities)" and lists four cities: Delhi, Kolkata, Istanbul, and Karachi. The table is a 4x4 grid of empty cells.

Items (Vehicle)	Delhi	Kolkata	Istanbul	Karachi
Car				
Bus				
Train				
Plane				



# **PHẦN 3: SLOWLY CHANGING DIMENSIONS**



# SLOWLY CHANGING DIMENSIONS

Slowly changing dimensions (SCD): là các dimension trong đó **dữ liệu thay đổi chậm rãi** theo thời gian thay vì thay đổi thường xuyên

SCD là một phần quan trọng trong việc quản lý dữ liệu và đảm bảo rằng hệ thống data warehouse có thể hiển thị chính xác thông tin theo thời gian.



# SLOWLY CHANGING DIMENSIONS

**TYPE 1: OVERWRITE  
THE EXISTING  
ATTRIBUTE**

—  
GHI ĐÈ LÊN DỮ  
LIỆU HIỆN CÓ

**TYPE 2: ADD A NEW  
DIMENSION ROW**

—  
THÊM HÀNG  
DIMENSION MỚI

**TYPE 3: ADD A NEW  
DIMENSION  
ATTRIBUTE**

—  
THÊM CỘT MỚI  
TRÊN CÙNG MỘT  
BẢNG GHI

**TYPE 6: KẾT HỢP  
1+2+3**

—



# TYPE 1 : OVERWRITE THE EXISTING ATTRIBUTE

**Ghi đè hoặc cập nhật trực tiếp giá trị cũ bằng giá trị mới** mà không tạo thành bản ghi mới hoặc lưu trữ bất kỳ thông tin nào về sự thay đổi.

**Thích hợp cho:**

- Sửa các lỗi sai của dữ liệu
- Thay đổi ở nơi các mối liên hệ lịch sử không còn quan trọng
- Giá trị cũ không còn ý nghĩa

Customer			
CustomerID	Custome_Name	Phone	City
1	Nguyễn Văn A	0124563789	Đà Nẵng
2	Lê Thị B	0245789456	Hải Phòng
3	Trần Văn Huy	0543986674	Huế



Customer			
CustomerID	Custome_Name	Phone	City
1	Nguyễn Văn A	0124563789	Hà Nội
2	Lê Thị B	0245789456	Hải Phòng
3	Trần Văn Huy	0543986674	Huế



## TYPE 2: ADD A NEW DIMENSION ROW

- **Phương pháp phổ biến nhất** vì nó cho phép **lưu trữ lịch sử** của dimension bằng cách thêm 1 bản ghi mới khi có sự thay đổi dữ liệu cộng với việc theo dõi thông tin về thời gian hiệu lực của mỗi bản ghi.

Customer							
STT	CustomerID	Custome_Name	Phone	City	IsActive	Active_Start	Active_End
1	1	Nguyễn Văn A	0124563789	Hà Nội	1	1/1/2020	9999999999
2	2	Lê Thị B	0245789456	Hà Nội	1	2/1/2020	9999999999



Customer							
STT	CustomerID	Custome_Name	Phone	City	IsActive	Active_Start	Active_End
1	1	Nguyễn Văn A	0124563789	Hà Nội	0	1/1/2020	9/9/2020
2	2	Lê Thị B	0245789456	Hà Nội	1	2/1/2020	9999999999
3	1	Nguyễn Văn A	0124563789	Hải Phòng	1	10/9/2020	9999999999



## TYPE 3: ADD A NEW DIMENSION ATTRIBUTE

- Phương pháp này **ít được sử dụng** và được dùng để lưu trữ lịch sử của dữ liệu, nhưng khác với SCD Loại 2 thì SCD Loại 3 chỉ lưu thông tin về sự thay đổi gần nhất và không theo dõi lịch sử thay đổi xảy ra trong quá khứ.
- Giá trị mới được lưu vào cột hiện có, giá trị cũ được lưu vào cột mới

Customer				
CustomerID	Custome_Name	Phone	City	Address_Prev
1	Nguyễn Văn A	0124563789	Hà Nội	
2	Lê Thị B	0245789456	Hải Phòng	
3	Trần Văn Huy	0543986674	Huế	

Customer				
CustomerID	Custome_Name	Phone	City	Address_Prev
1	Nguyễn Văn A	0124563789	Cà Mau	Hà Nội
2	Lê Thị B	0245789456	Hải Phòng	
3	Trần Văn Huy	0543986674	Huế	



## TYPE 6: KẾT HỢP 1+2+3

Kết hợp SCD Type 1, SCD Type 2 và SCD Type 3. Giá trị mới ghi đè lên giá trị cũ nhưng **giá trị cũ vẫn được lưu trữ** trong bảng chính, **cùng với giá trị hiện tại**.

Customer								
STT	CustomerID	Custome_Name	Phone	City	City_Prev	Is_Active	Active_Start	Active_End
1	1	Nguyễn Văn A	0124563789	Đà Nẵng	Đà Nẵng	1	1/1/2020	9999999999
Customer								
STT	CustomerID	Custome_Name	Phone	City	City_Prev	Is_Active	Active_Start	Active_End
1	1	Nguyễn Văn A	0124563789	Hải Phòng	Đà Nẵng	0	1/1/2020	9/10/2020
2	1	Nguyễn Văn A	0124563790	Hải Phòng	Hải Phòng	1	10/1/2020	9999999999
Customer								
STT	CustomerID	Custome_Name	Phone	City	City_Prev	Is_Active	Active_Start	Active_End
1	1	Nguyễn Văn A	0124563789	Hà Nội	Đà Nẵng	0	1/1/2020	9/10/2020
2	1	Nguyễn Văn A	0124563789	Hà Nội	Hải Phòng	0	10/1/2020	19/10/2022
3	1	Nguyễn Văn A	0124563789	Hà Nội	Hà Nội	1	20/10/2022	9999999999



# **PHẦN 4: RAPIDLY CHANGING DIMENSIONS**







# RAPIDLY CHANGING DIMENSIONS

- Rapidly Changing Dimensions (RCD) là các dimension có các thuộc tính trong đó các giá trị sẽ được **thay đổi thường xuyên**.
- Việc thay đổi giá trị thường xuyên dẫn đến **kích thước** của bảng dimension **tăng nhanh** --> nhiều vấn đề nghiêm trọng về hiệu suất trong mô hình dữ liệu
- "**Mini-Dimension**" là một giải pháp được sử dụng để giải quyết Rapidly Changing Dimensions (RCD) giúp làm **giảm kích thước** bảng dữ liệu đồng thời **cải thiện hiệu suất truy vấn**







**CUST\_DIM**

 CUST\_KEY  
 START\_DATE  
 END\_DATE  
 CUST\_NAME  
 DATE\_OF\_BIRTH  
 CUST\_STATE  
 ...

**SALES\_FACT**

 CUST\_KEY (FK)  
 CUST\_DEMO\_KEY (FK)  
 DATE\_KEY (FK)  
 PROD\_KEY (FK)  
 ...  
 ...  
 ...

**CUST\_DEMO\_DIM**

 CUST\_DEMO\_KEY  
 AGE\_BAND  
 INCOME\_BAND  
 PURCHASE\_BAND





Thanks!

