

---

# COMP9318: Data Warehousing and Data Mining

Assignment Project Exam Help

— L2: <https://eduassistpro.github.io/> OLAP

Add WeChat edu\_assist\_pro

---

- Why and What are Data Warehouses?

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# Data Analysis Problems

---

- The same data found in many different systems
  - Example: customer data across different departments
- Assignment Project Exam Help
- The same https://eduassistpro.github.io/ifferently
- Heterogeneous
  - Relational DBMS, Online Processing (OLTP)
  - Unstructured data in files (e.g., MS Excel) and documents (e.g., MS Word)

# Data Analysis Problems (Cont'd)

---

- Data is suited for operational systems
  - Accounting, billing, etc.
  - Do not support analysis across business functions
- Data quality is
  - Missing data, imprecise
- Data are “volatile”
  - Data deleted in operational systems (6months)
  - Data change over time – no historical information

# Solution: Data Warehouse

---

- Defined in many different ways, but not rigorously.
  - A decision support database that is maintained **separately** from the organization's operational database  
**Assignment Project Exam Help**
  - Support **inform** g a solid platform of consolidated, h  
<https://eduassistpro.github.io/>
- "A data warehouse **Adds** **Subject** **edu\_assist\_pro**, **one-variant**, and **nonvolatile** collection of data in support of management's decision-making process."—W. H. Inmon
- Data warehousing:
  - The process of constructing and using data warehouses

# Data Warehouse—Subject-Oriented

---

- Organized around major subjects, such as **customer**, **product**, **sales**.
- Focusing on the modeling and analysis of data for decision makers, not on <https://eduassistpro.github.io/>
- Provide a simple and concise **Add WeChat edu\_assist\_pro** d particular subject issues by excluding data that are not useful in the decision support process.

# Data Warehouse—Integrated

---

- Constructed by integrating multiple, heterogeneous data sources
  - relational databases, flat files, on-line transaction records
- Data cleaning <https://eduassistpro.github.io/> techniques are applied.
  - Ensure consistency in names, types, encoding structures, attribute measures, etc. among different data sources
    - E.g., Hotel price: currency, tax, breakfast covered, etc.
    - When data is moved to the warehouse, it is converted.

# Data Warehouse—Time Variant

---

- The time horizon for the data warehouse is significantly longer than that of operational systems.
  - Operational database current value data.
  - Data warehouse information from a historical perspective (<https://eduassistpro.github.io/>)
- Every key structure in the data
  - Contains an element of time, explicitly or implicitly
  - But the key of operational data may or may not contain “time element”.

# Data Warehouse—Non-Volatile

---

1. A physically separate store of data transformed from the operational environment.
2. Operational ~~Assistants of Data~~ ~~Explain Help~~ in the data warehouse en <https://eduassistpro.github.io/>
  - Does not require transaction, recovery, and concurrency control
  - Requires only two operations in data accessing:
    - *initial loading of data* and *access of data*.

# Data Warehouse Architecture

---

- Extract data from operational data sources
  - clean, transform
- Bulk load/refresh
- warehouse is of <https://eduassistpro.github.io/>
- OLAP-server prov multidimensional view
- Multidimensional-olap
  - (Essbase, oracle express)
- Relational-olap
  - (Redbrick, Informix, Sybase, SQL server)

Assignment Project Exam Help

Add WeChat edu\_assist\_pro

(Essbase, oracle  
express)

Relational-olap  
(Redbrick, Informix,  
Sybase, SQL server)

# Data Warehouse Architecture

---

All subjects,  
integrated

Advanced  
analysis

Assignment Project Exam Help

 <https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

Function-oriented  
systems

Subject-oriented  
systems

# Why Separate Data Warehouse?

---

- High performance for both systems
  - DBMS— tuned for OLTP: access methods, indexing, concurrency control, recovery
  - Warehouse— tuned for P queries, multidimensional analysis
- Different functions and different data
  - missing data: Decision support data which operational DBs do not typically maintain
  - data consolidation: DS requires consolidation (aggregation, summarization) of data from heterogeneous sources
  - data quality: different sources typically use inconsistent data representations, codes and formats which have to be reconciled

**Assignment Project Exam Help**

**Add WeChat [edu\\_assist\\_pro](https://eduassistpro.github.io/)**

# Why OLAP Servers?

---

- Different workload:
  - OLTP (on-line transaction processing)
    - Major task of traditional relational DBMS
    - Day-to-day operations: purchasing, inventory, banking, manufacturing, payroll, registration, accounting, etc.
  - OLAP (on-line <https://eduassistpro.github.io/>)
    - Major task of
    - Data analysis and decision-making
- Queries hard/infeasible for OLTP, e.g.,
  - Which **week** we have the largest sales?
  - Does the sales of **dairy products** increase over time?
  - Generate a **spread sheet** of total sales by state and by year.
- Difficult to represent these queries by using SQL ← Why?

# OLTP vs. OLAP

	<b>OLTP</b>	<b>OLAP</b>
<b>users</b>	clerk, IT professional	knowledge worker
<b>function</b>	day to day operations	decision support
<b>DB design</b>	application oriented	subject oriented
<b>data</b>	current, up-to-date det iso	historical, zied, multidimensional d, consolidated
<b>usage</b>	repetitive	
<b>access</b>	read/write index/hash on prim. key	
<b>unit of work</b>	short, simple transaction	complex query
<b># records accessed</b>	tens	millions
<b>#users</b>	thousands	hundreds
<b>DB size</b>	100MB-GB	100GB-TB
<b>metric</b>	transaction throughput	query throughput, response

# Comparisons

	Databases	Data Warehouses
Purpose	Many purposes; Flexible and general	One purpose: Data analysis
Conceptual Model	ER	Multidimensional
Logical Model	(Normalized) Relational	Normalized) Star schema / cube/cuboids <a href="https://eduassistpro.github.io/">https://eduassistpro.github.io/</a> Add WeChat edu_assist_pro
Physical Model	Relational Tables	ROLAP: Relational tables MOLAP: Multidimensional arrays
Query Language	SQL (hard for analytical queries)	MDX (easier for analytical queries)
Query Processing	B+-tree/hash indexes, Multiple join optimization, Materialized views	Bitmap/Join indexes, Star join, Materialized data cube

---

## ■ The Multidimensional Model

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# The Multidimensional Model

---

- A data warehouse is based on a multidimensional data model which views data in the form of a **data cube**, which is a multidimensional generalization of 2D spread sheet.
- Key concepts:  
  - **Facts:** the s
    - Typically tr <https://eduassistpro.github.io/> er types includes snapshots, etc.
    - Measures: numbers that can d
    - Dimensions: context of the measure
  - Hierarchies:
    - Provide contexts of different granularities (aka. grains)
- Goals for dimensional modeling:
  - Surround facts with as much relevant context (dimensions) as possible ← Why?

# Supermarket Example

---

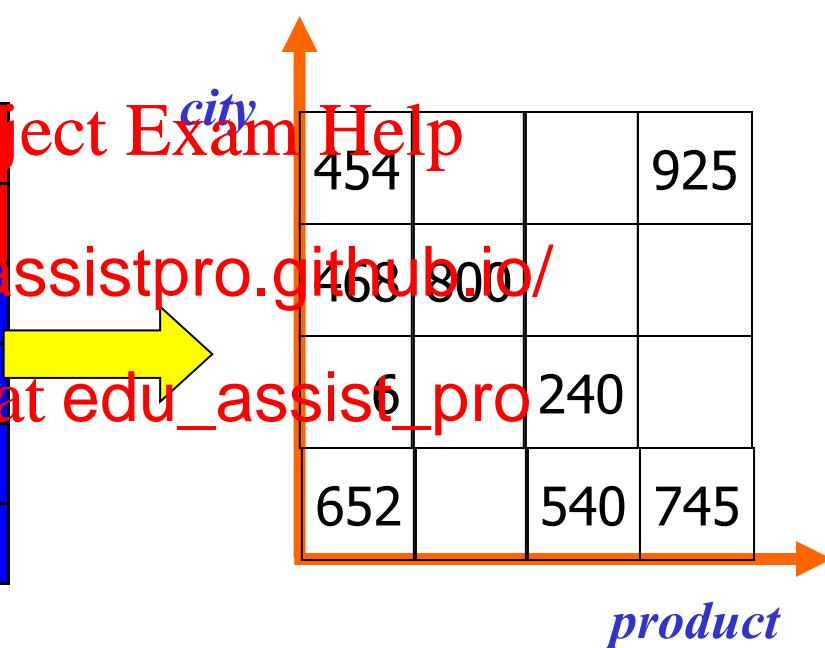
- Subject: analyze total sales and profits
- Fact: Each Sales **Transaction**
  - Measure: Dollars\_Sold, Amount\_Sold, Cost
  - Calculated M
- Dimensions: <https://eduassistpro.github.io/>
  - Store
  - Product
  - Time

Add WeChat edu\_assist\_pro

# Visualizing the Cubes

- A valid **instance** of the model is a data cube

total Sales		product			
city	NY	p1	p2	-	-
	LA	\$454	-		
	SD	\$468	\$800		
	SF	\$296	-	\$240	-
		\$652	-	\$540	\$745



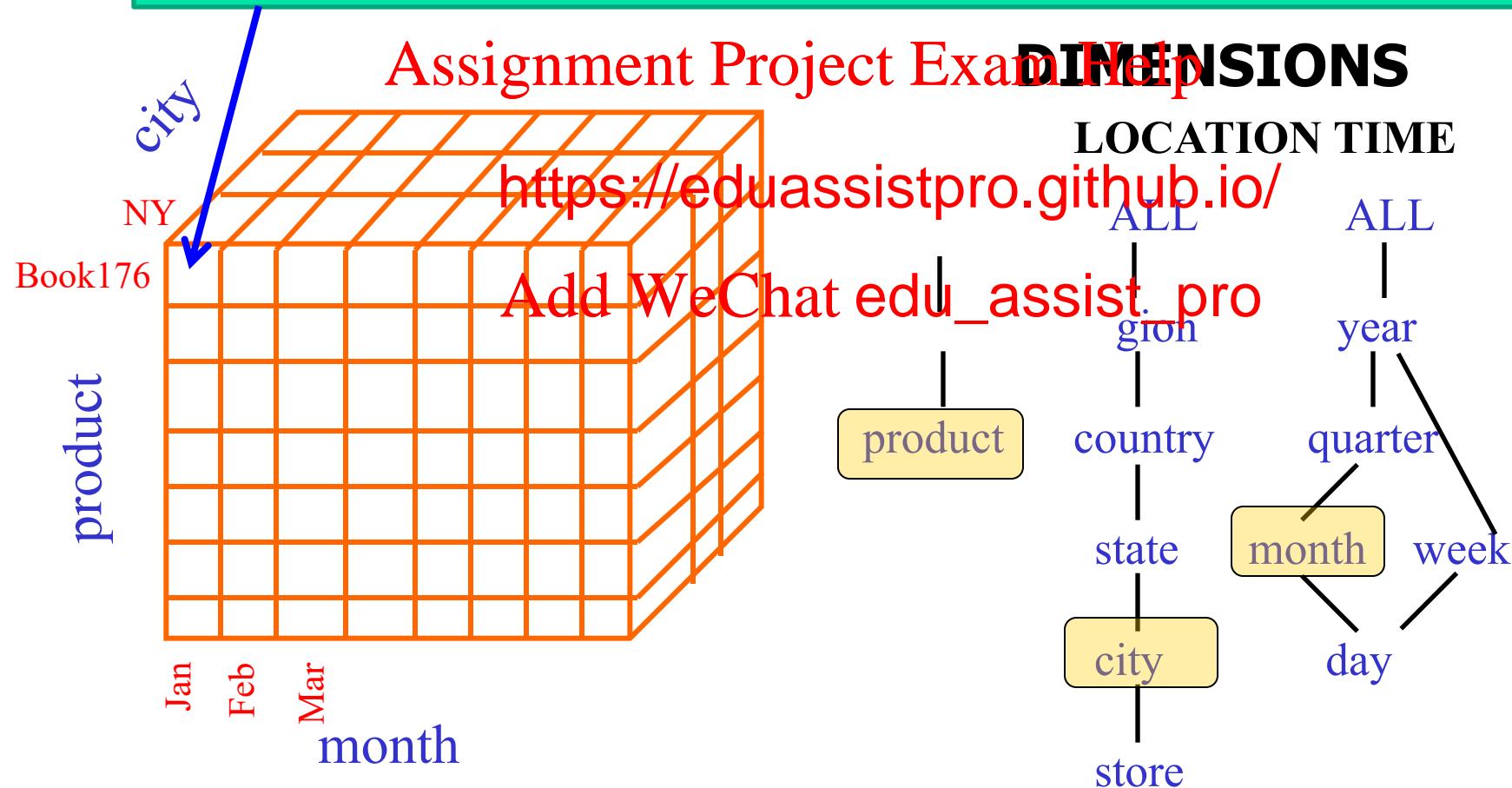
**Concepts:** cell, fact (=non-empty cell), measure, dimensions

Q: How to generalize it to 3D?

# 3D Cube and Hierarchies

**Concepts:** hierarchy (a tree of dimension values), level

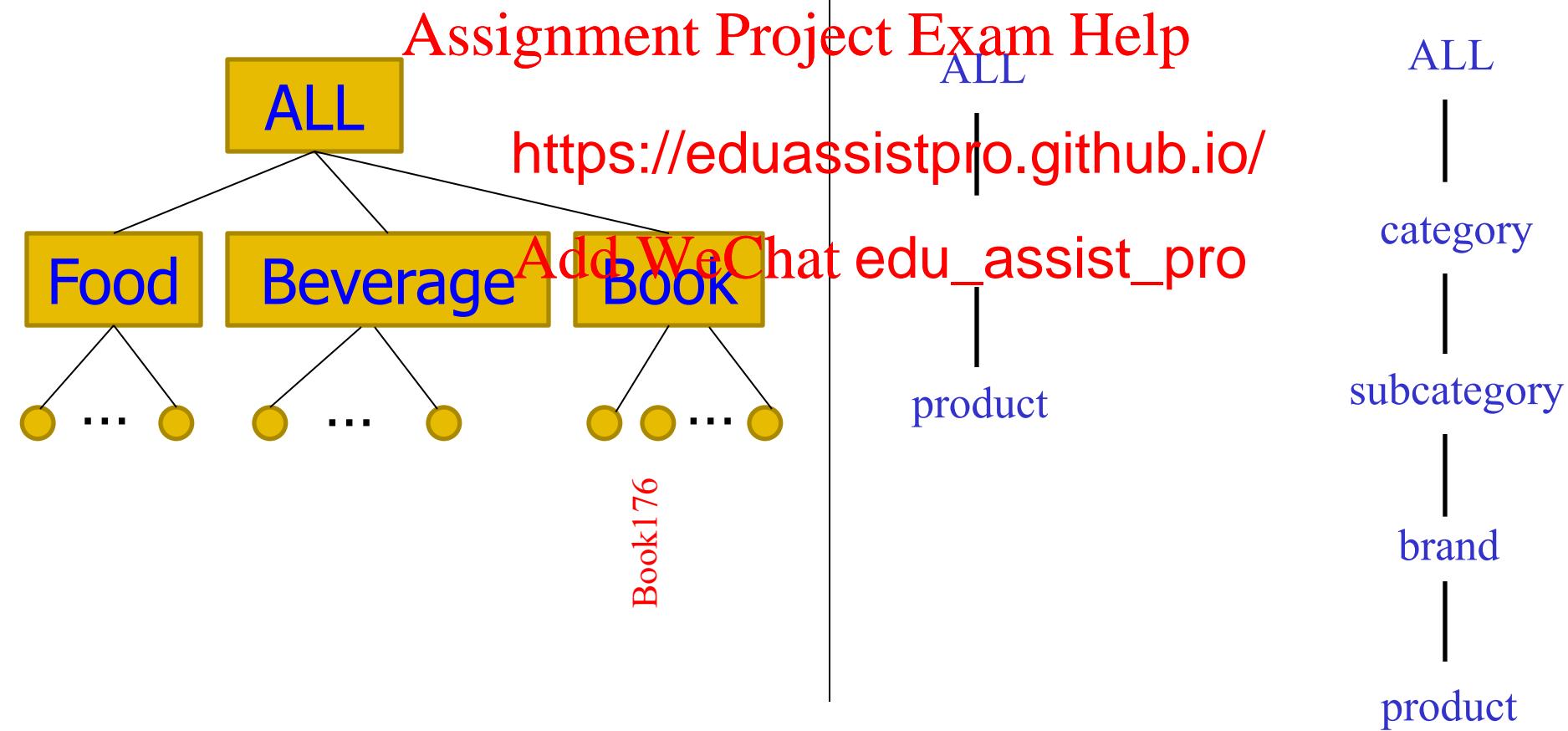
Sales of book176 in NY in Jan can be found in this cell



# Hierarchies

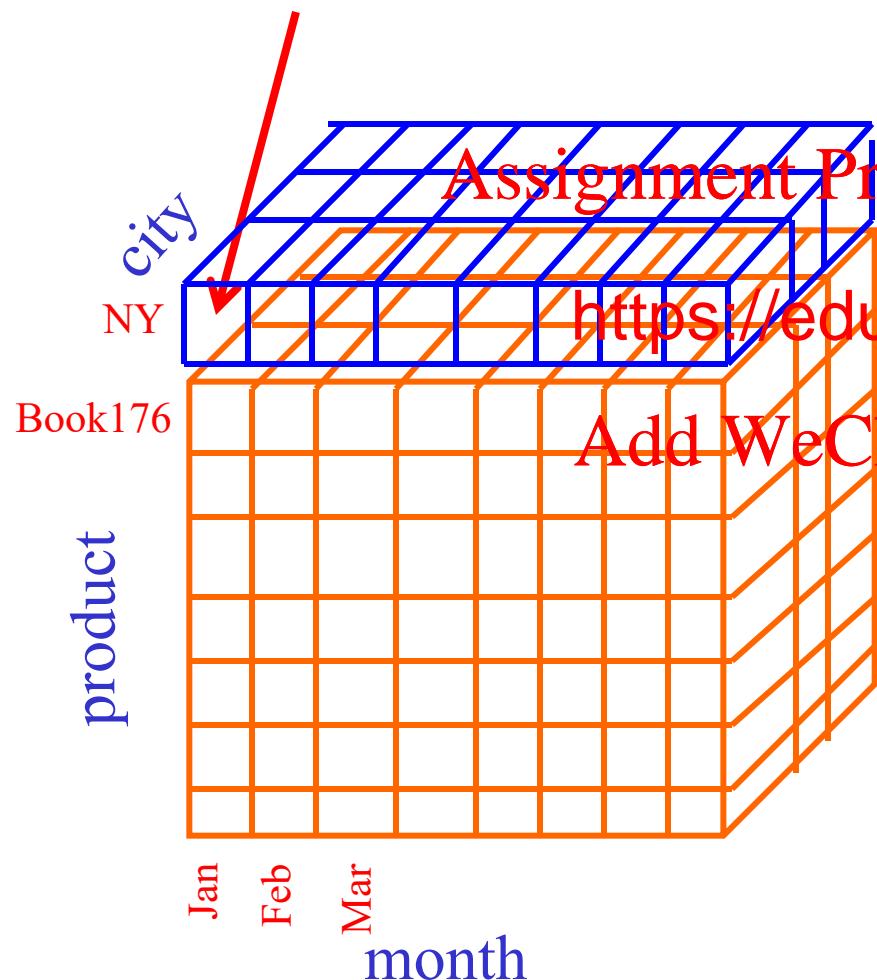
**Concepts:** hierarchy (a tree of dimension values), level

Which design is better? Why?



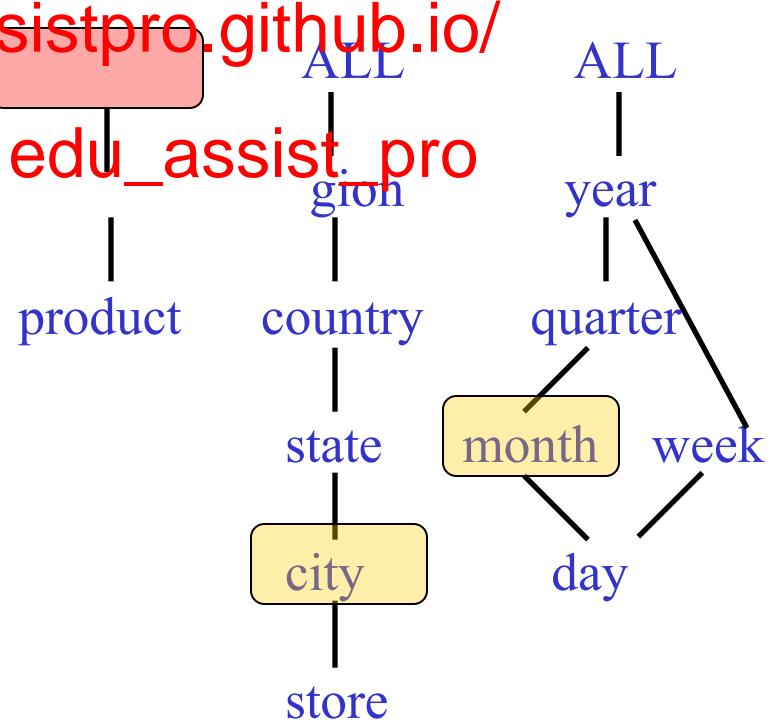
# The (city, moth) Cuboid

Sales of ALL\_PROD in NY in Jan



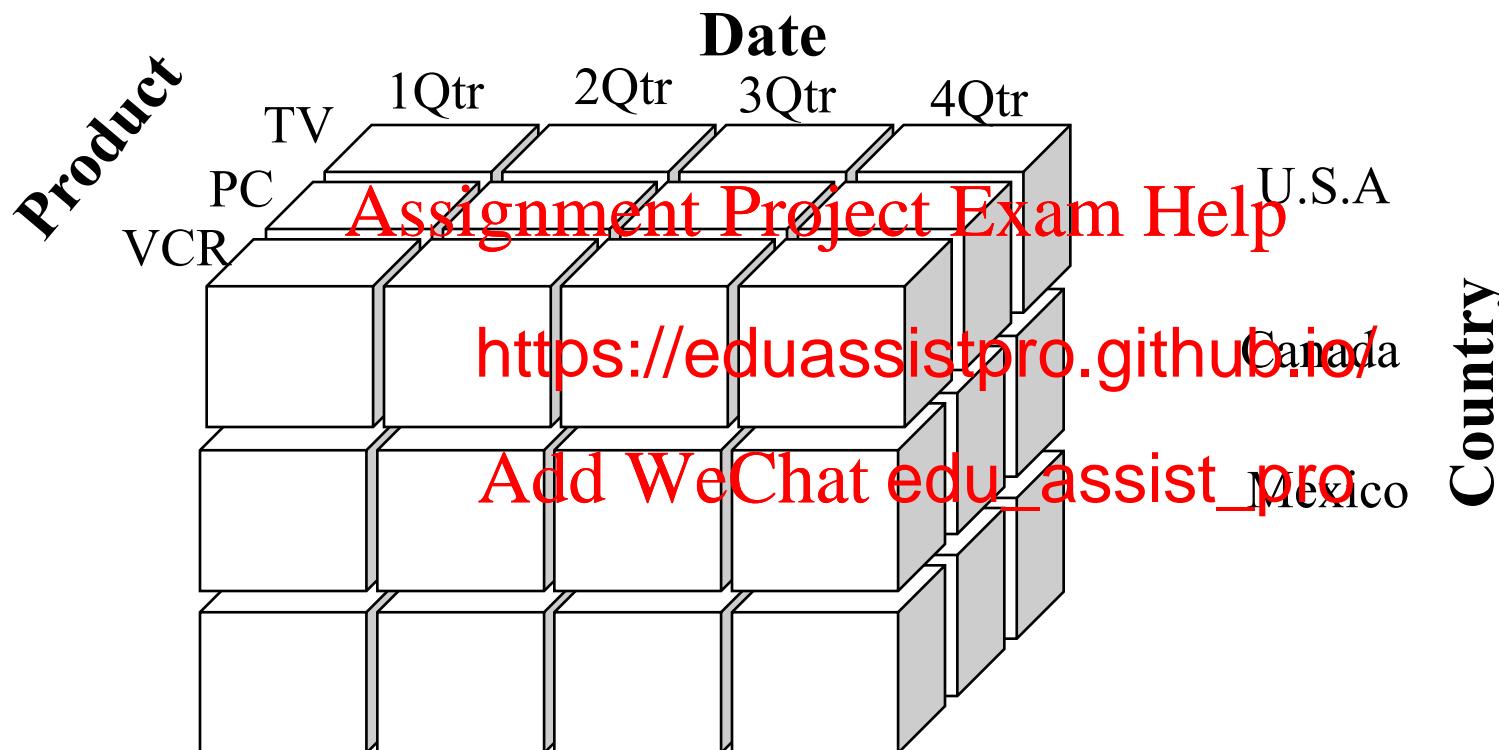
Assignment Project Exam **DIMENSIONS**

LOCATION TIME



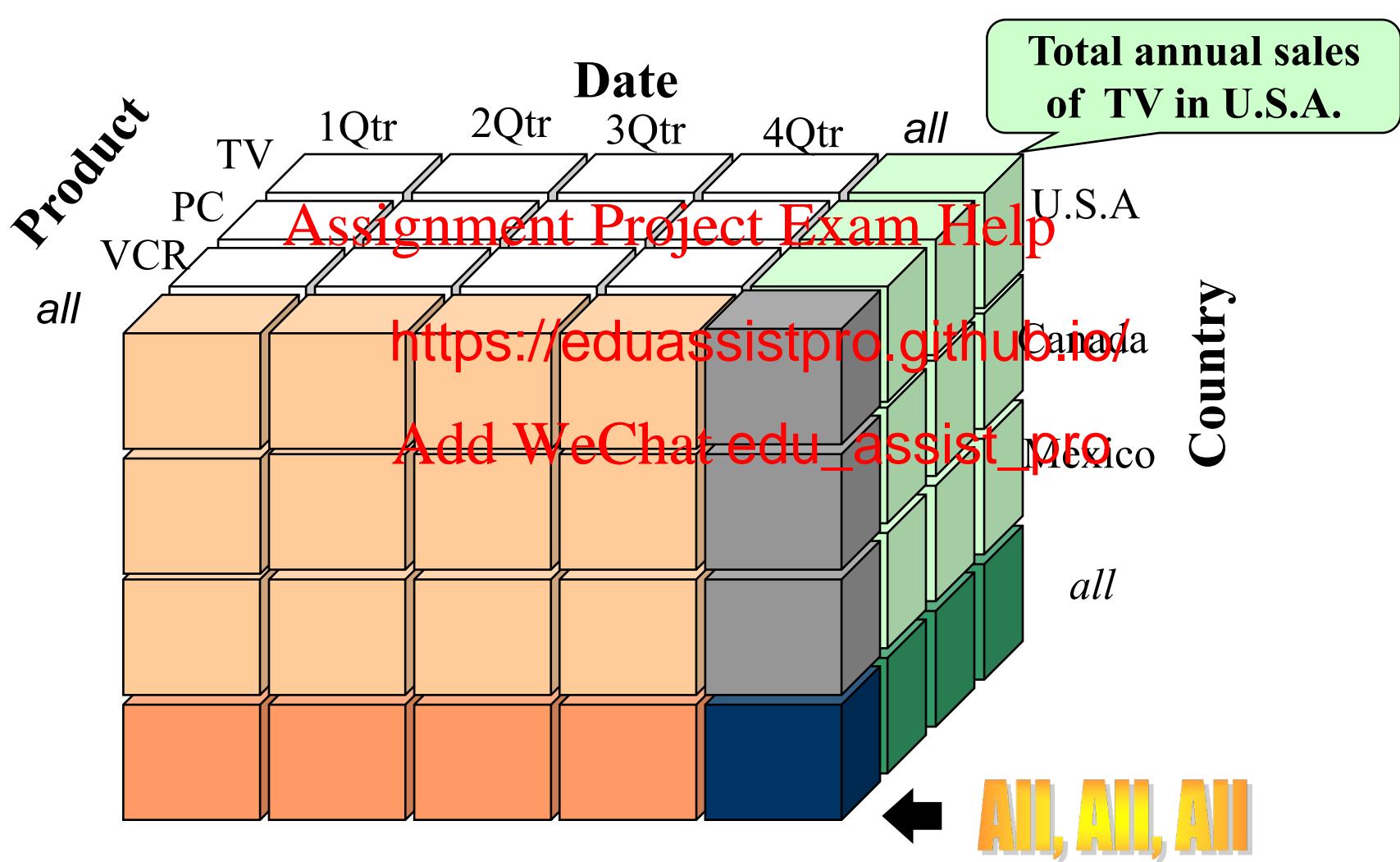
# All the Cuboids

Assume: no other non-ALL levels on all dimensions.

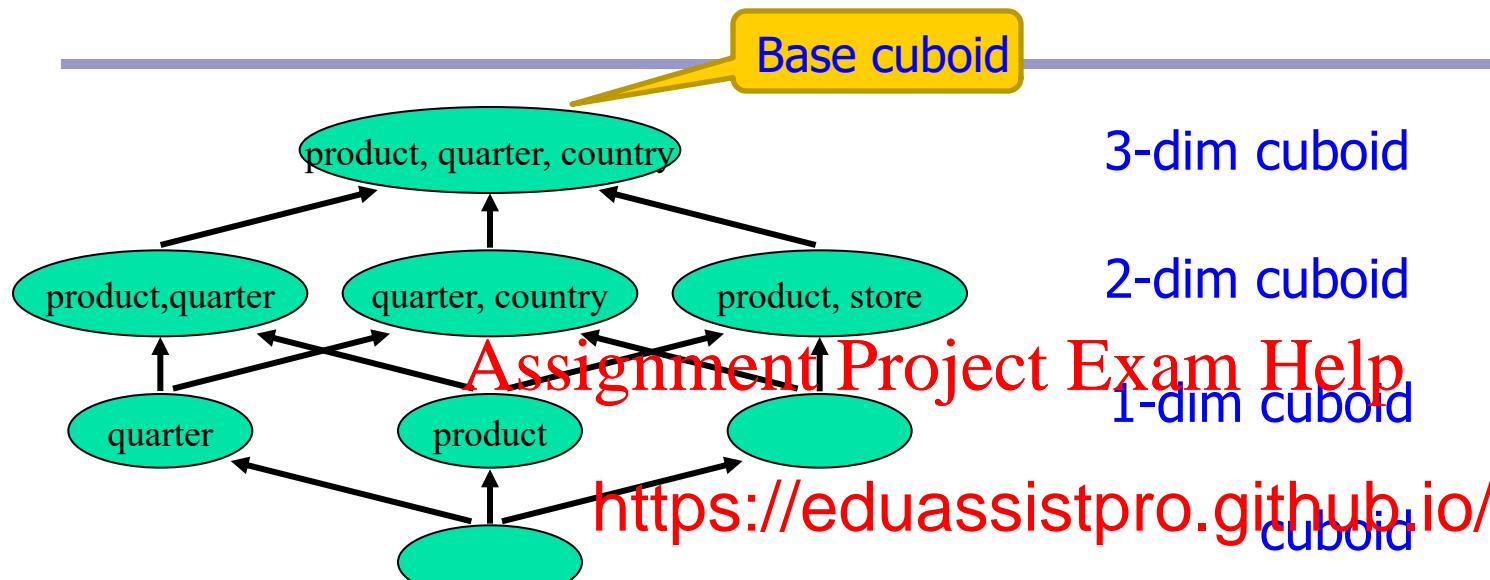


## All the Cuboids /2

Assume: no other non-ALL levels on all dimensions.



# Lattice of the cuboids



- n-dim cube can be represented as  $\prod_{i=1}^d D_i$ , where  $D_i$  is the set of allowed values on the i-th dimension
  - if  $D_i = L_i$  (a particular level), then  $D_i = \text{all descendant dimension values of } L_i$ .
  - ALL can be omitted and hence reduces the effective dimensionality
- A complete cube of d-dimensions consists of  $\prod_{i=1}^d (n_i + 1)$  cuboids, where  $n_i$  is the number of levels (excluding ALL) on i-th dimension.
  - They collectively form a lattice.

# Properties of Operations

---

- All operations are closed under the multidimensional model
  - i.e., both ~~Assignment Project Exam Help~~ operation is a cube
- So that they <https://eduassistpro.github.io/>  
[Add WeChat edu\\_assist\\_pro](#)

Q: What's the analogy in the Relational Model?

# Common OLAP Operations

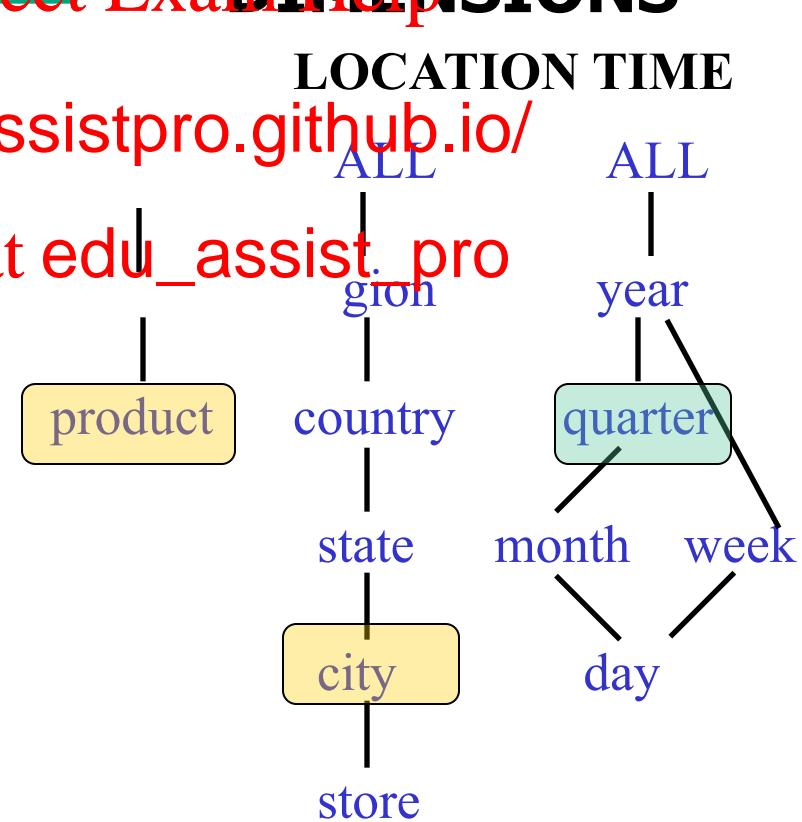
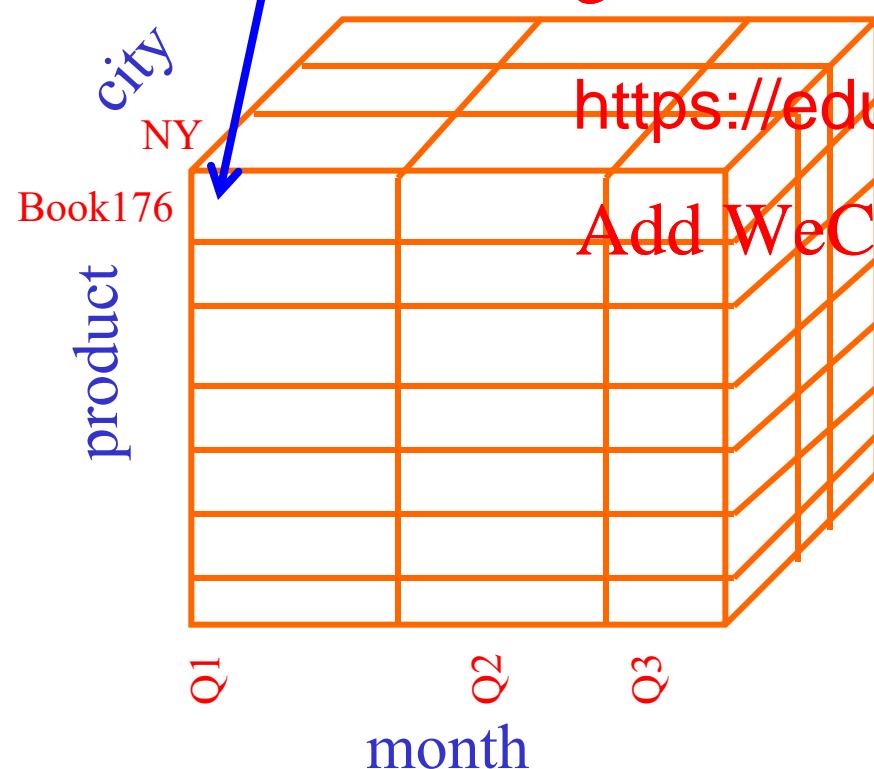
- **Roll-up:** move up the hierarchy

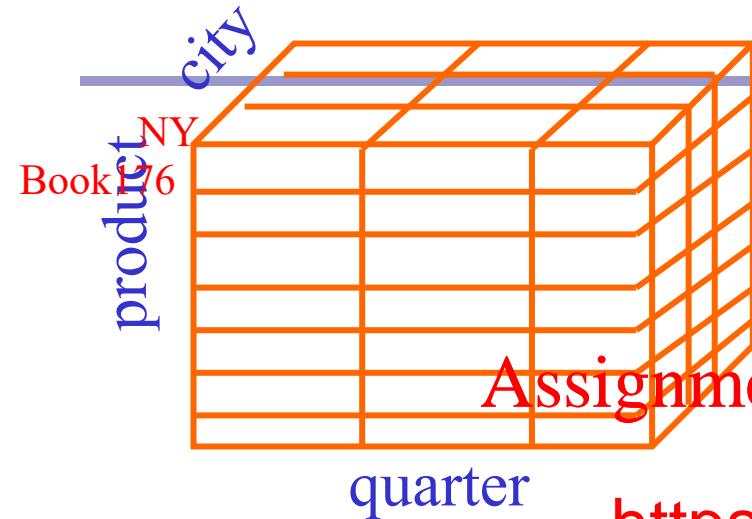
Q: what should be its value?

Sales of book176 in NY in Q1 here

Assignment Project Exam

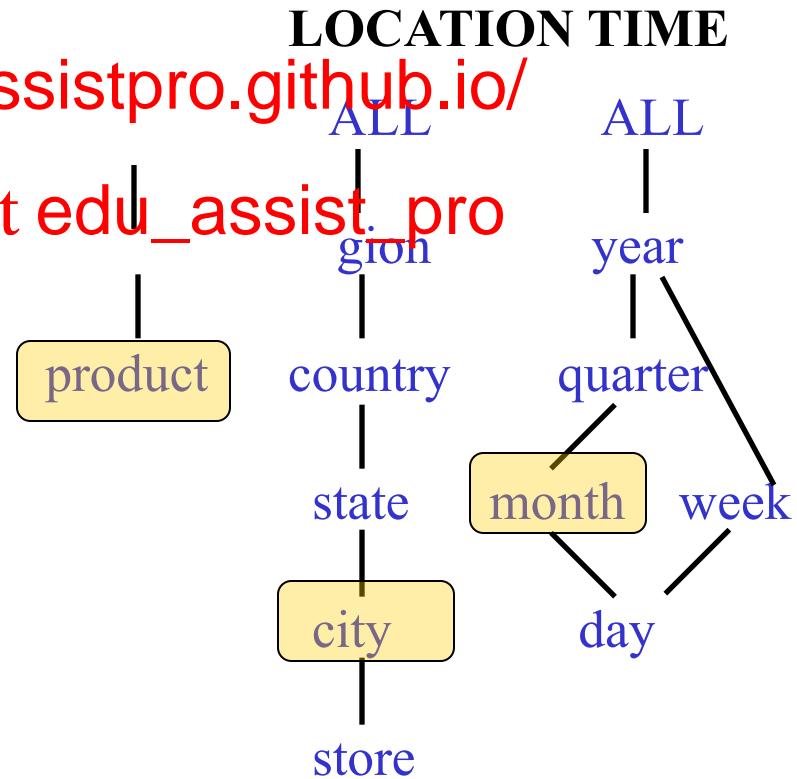
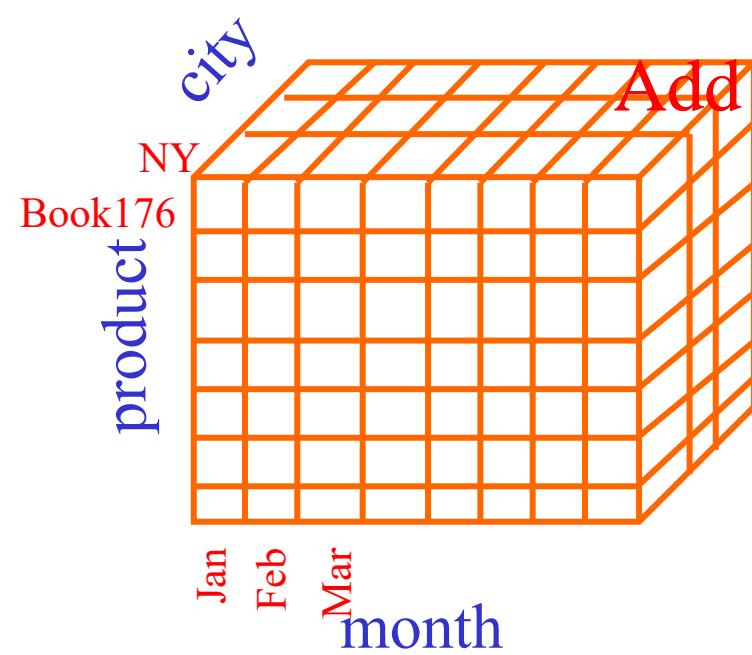
**DIMENSIONS**





Assignment Project Exam **DIMENSIONS**

<https://eduassistpro.github.io/>



# Data Cube Measures: Three Categories

---

- **Distributive**: if the result derived by applying the function to  $n$  aggregate values is the same as that derived by applying the function on all the data without partitioning
  - E.g., count(), sum(), min(), max()
- **Algebraic**: if it contains <https://eduassistpro.github.io/> function with  $M$  arguments (where  $M$  is integer), each of which is obtained by applying aggregate function
  - E.g., avg(), min\_N(), standard\_deviation()
- **Holistic**: if there is no constant bound on the storage size needed to describe a subaggregate.
  - E.g., median(), mode(), rank()

# Common OLAP Operations

---

- **Drill-down:** move down the hierarchy

- more fine-grained aggregation

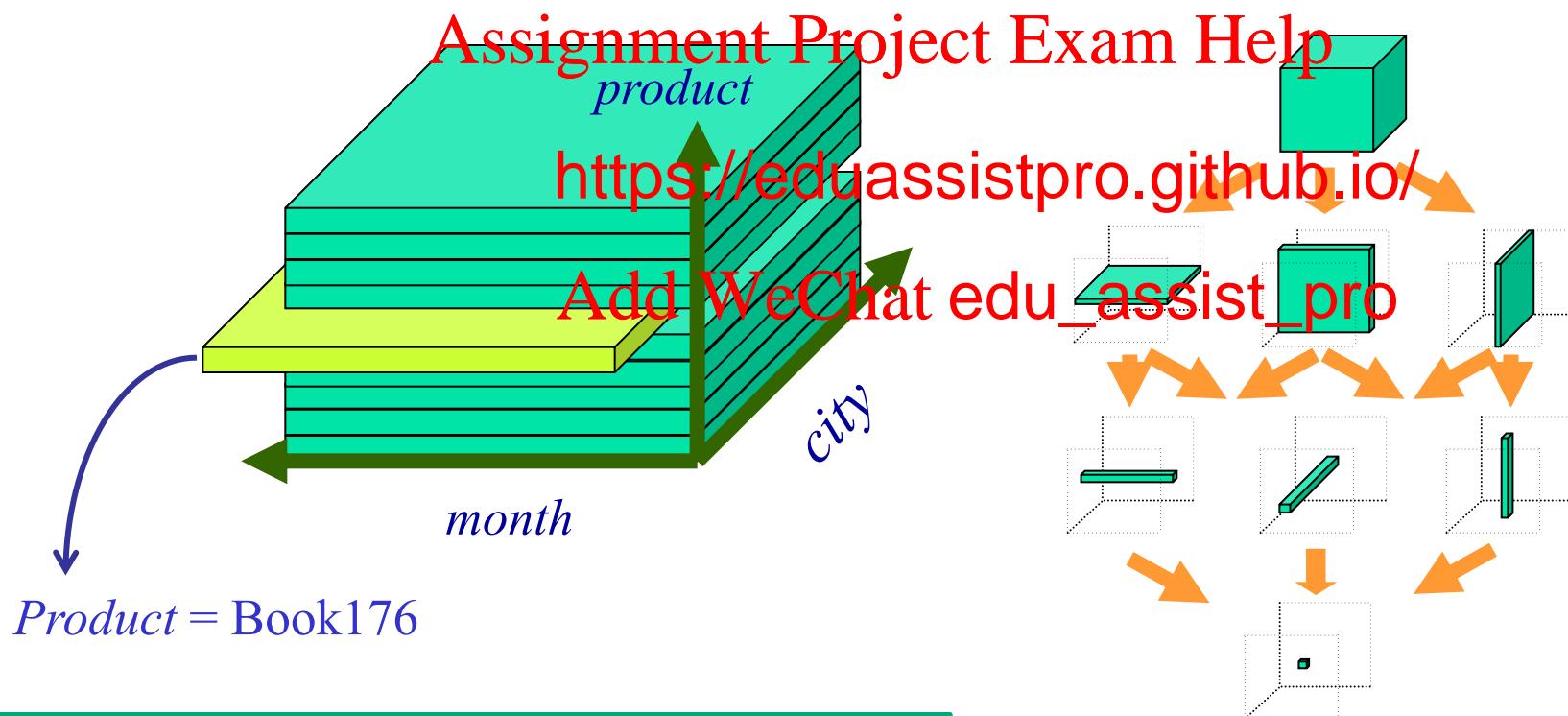
Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# Slice and Dice Queries

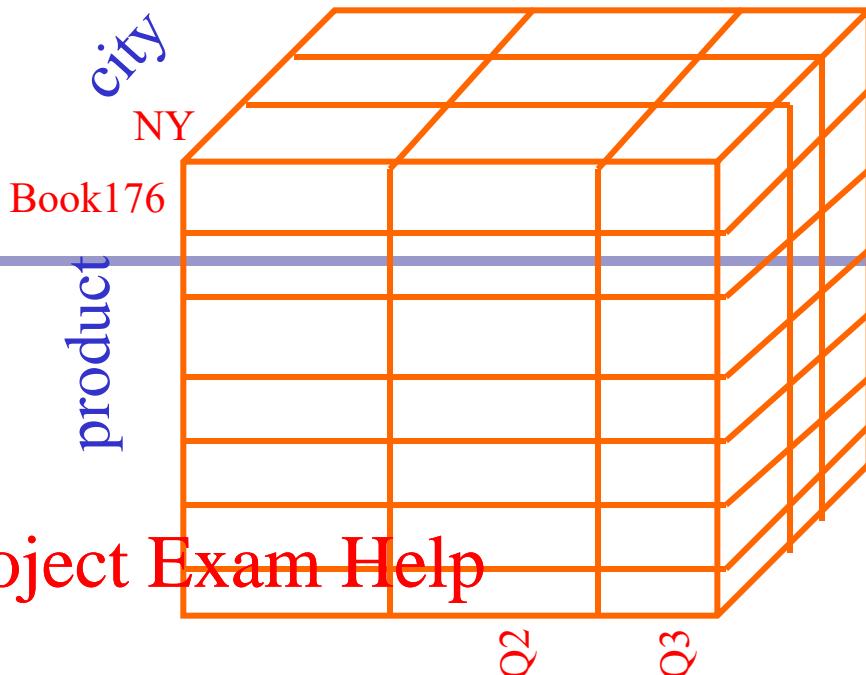
- Slice and Dice: select and project on one or more dimension values



The output cube has smaller dimensionality than the input cube

# Pivoting

- Pivoting: aggregate on selected dimensions
  - usually 2 dims (cross-tabulation)



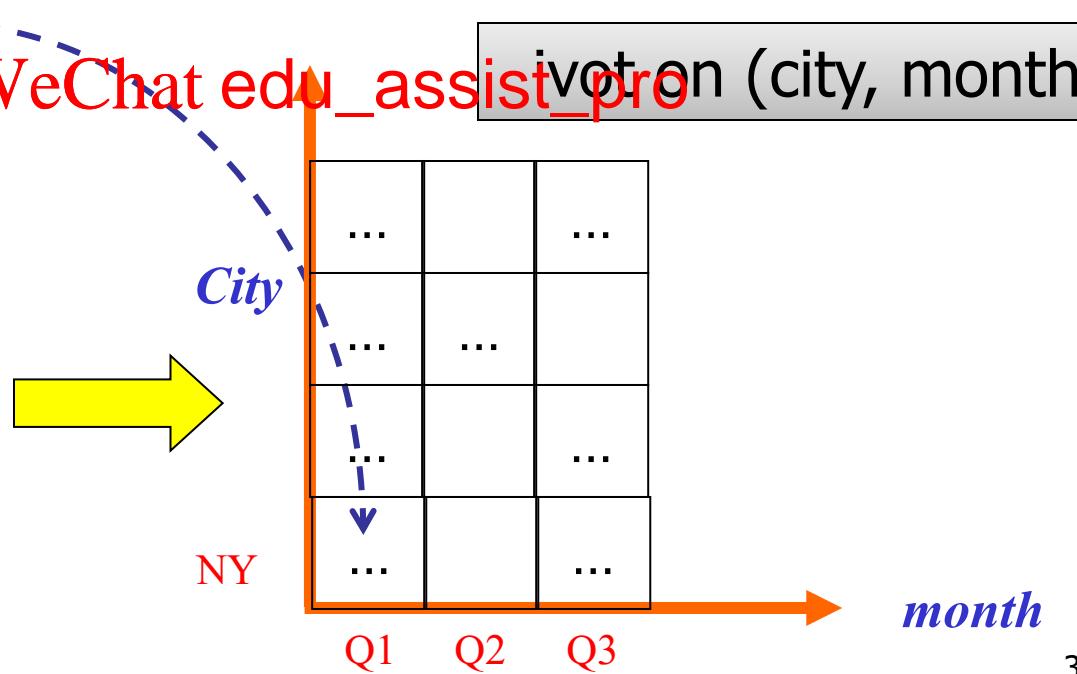
Sales (of all products) in NY in Q1

=sum( ??? )

Add WeChat edu\_assist\_pro

<https://eduassistpro.github.io/>

pivot on (city, month)



# A Reflective Pause

---

- Let's review the definition of data cubes again.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

- **Key message:**
  - Disentangle the “object” from its “representation” or “implementation”

# Modeling Exercise 1: Monthly Phone Service Billing

---

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

Theme: analyze the income/revenue of Telstra

# Solution

---

- FACT

Assignment Project Exam Help

- MEASURE <https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

- DIMENSIONS

---

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

---

- The Logical Model

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# Logical Models

---

- Two main approaches:
    - Using relational DB technology:
      - Star schema, Snowflake schema, Fact constellation
    - Using multi-model: <https://eduassistpro.github.io/>
      - Just as me
- Add WeChat edu\_assist\_pro

# Universal Schema → Star Schema

- Many data warehouses adopt a star schema to represent the multidimensional model
- Each dimension is represented by a dimension-table
  - LOCATION (location\_key, store, street, address, city, state, country, region)
  - dimension tabl <https://eduassistpro.github.io/>
- Transactions are described th t-table
  - each tuple consists of a logical p Add WeChat [edu\\_assist\\_pro](http://edu_assist_pro) or the dimension-tables (foreign-key) and a list of measures (e.g. sales \$\$\$)

The universal schema for supermarket

Store	City	State	Prod	Brand	Category	\$Sold	#Sold	Cost
S136	Syd	NSW	76Ha	Nestle	Biscuit	40	10	18
S173	Melb	Vic	76Ha	Nestle	Biscuit	20	5	11

# The Star Schema

TIME
time_key
day
day_of_the_week
month
quarter
year

SALES
location_key
units_sold
amount

PRODUCT
product_key
product_name
category
brand
color
supplier_name

LOCATION
location_key
store
street_address
city
state
country
region

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat [edu\\_assist\\_pro](https://edu-assist-pro)

Think why:

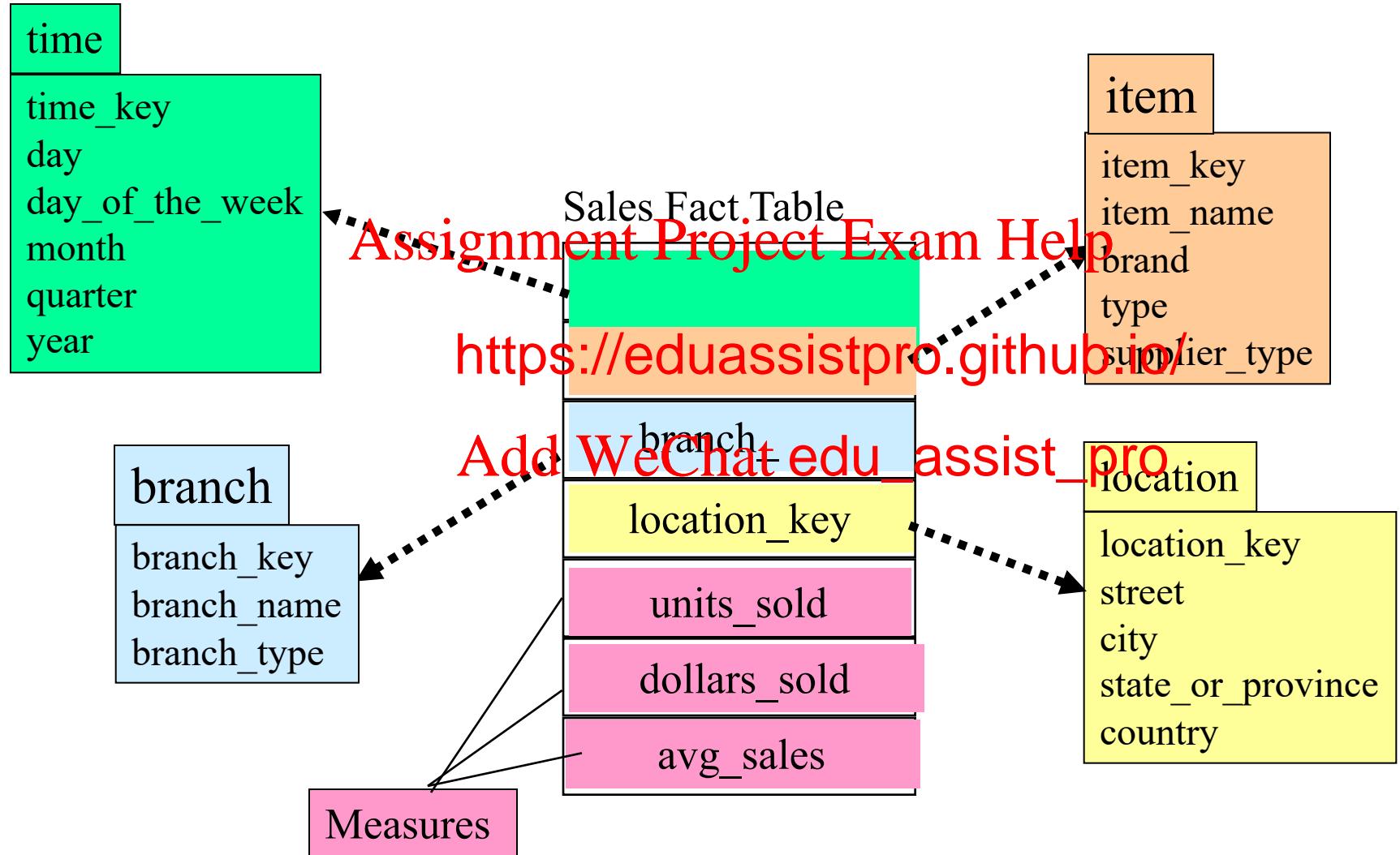
- (1) Denormalized **once** from the universal schema
- (2) Controlled **redundancy**

# Typical Models for Data Warehouses

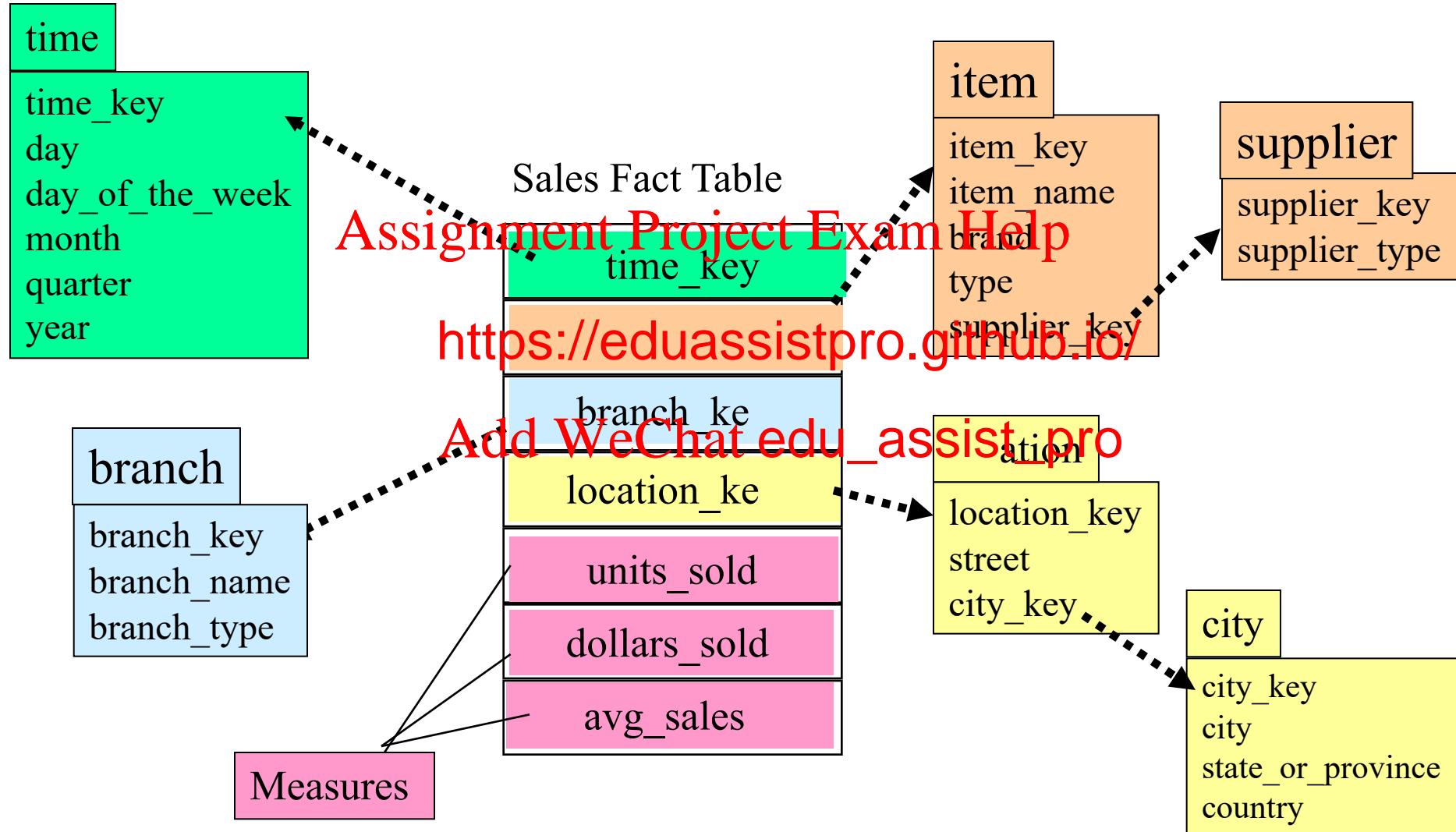
---

- Modeling data warehouses: dimensions & measures
  - Star schema: A fact table in the middle connected to a set of dimension tables
  - Snowflake schema: https://eduassistpro.github.io/  
where some dimensional hierarchy is normalized into a set of smaller dimension tables, resulting in a star shape similar to snowflake
  - Fact constellations: Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called **galaxy schema** or fact constellation

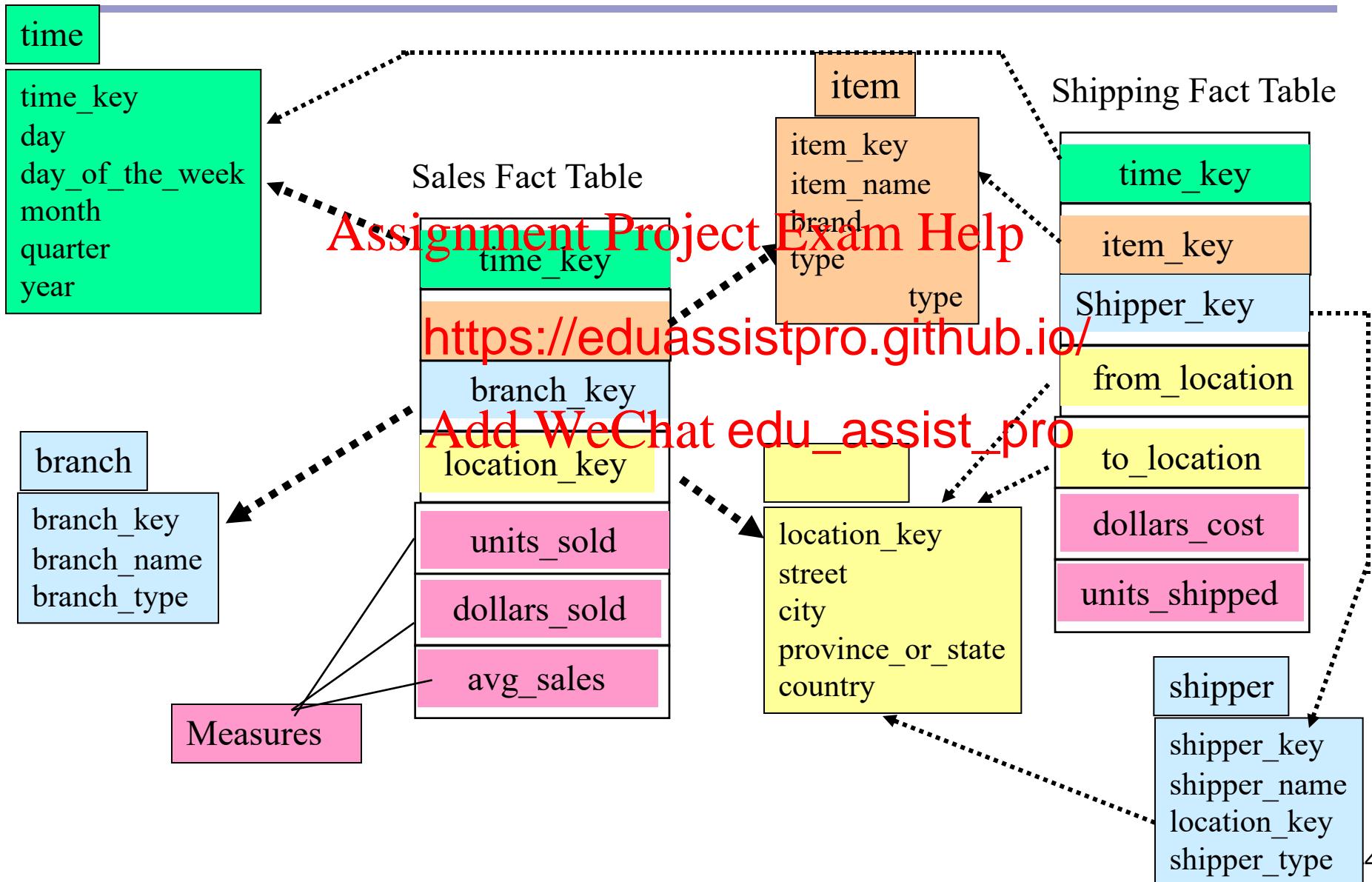
# Example of Star Schema



# Example of Snowflake Schema



# Example of Fact Constellation



# Advantages of Star Schema

---

- Facts and dimensions are clearly depicted
  - dimension tables are relatively static, data is loaded ~~(Assignment Project Exam Help)~~
  - easy to co [queries](https://eduassistpro.github.io/)

*“Find total sales per product category res in Europe”*  
[Add WeChat edu\\_assist\\_pro](https://eduassistpro.github.io/)

```
SELECT PRODUCT.category, SUM(SALES.amount)
FROM     SALES, PRODUCT, LOCATION
WHERE   SALES.product_key = PRODUCT.product_key
AND      SALES.location_key = LOCATION.location_key
AND      LOCATION.region = "Europe"
GROUP BY PRODUCT.category
```

Operations: Slice (Loc.Region.Europe) + Pivot (Prod.category)

---

- Query Language

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# Query Language

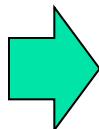
- Two approaches:
  - Using relational DB technology: SQL (with extensions such as CUBE/PIVOT/UNPIVOT)
  - Using multid

Assignment Project Exam Help  
MDX

<https://eduassistpro.github.io/>

```
SELECT PRODUCT.categ
SUM(SALES.amount)
FROM   SALES, PRODUCT, LOCATION
WHERE  SALES.product_key =
PRODUCT.product_key
AND    SALES.location_key =
LOCATION.location_key
AND    LOCATION.region="Europe"
GROUP BY PRODUCT.category
```

Add WeChat [edu\\_assist\\_pro](#)  
[category] on ROWS,  
{[MEASURES].[amount]} on COLUMNS  
FROM [SALES]  
WHERE ([LOCATION].[region].[Europe])



---

- Physical Model + Query Processing Techniques

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# Physical Model + Query Processing Techniques

---

- Two main approaches:
  - Using relational DB technology: ROLAP
  - Using ~~Assignment Project Exam Help~~: MOLAP
- Hybrid: HOL <https://eduassistpro.github.io/>
  - Base cuboid: ROLAP  
[Add WeChat edu\\_assist\\_pro](#)
  - Other cuboids: MOLAP

# Q1: Selection on low-cardinality attributes

TIME
time_key
day
day_of_the_week
month
quarter
year

$S_{region='Europe'}$

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat `edu_assist_pro`

customer\_k

location\_key

*units\_sold*

*amount*

CUSTOMER
customer_key
customer_name
region
type

LOCATION
location_key
store
street_address
city
state
country
region

- Ignoring the final GROUP BY for now
- Omitting the Product dimension

# Indexing OLAP Data: Bitmap Index

---

## (1) BI on dimension tables

- Index on an attribute (column) with low distinct values
  - Each distinct values, v, is associated with a n-bit vector (n = #rows)
    - The  $i$ -th bit is set if the  $i$ -th row of the table has the value v for the index
  - Multiple BIs can enable optimized scan of the table
- Assignment Project Exam Help  
<https://eduassistpro.github.io/>  
 Add WeChat edu\_assist\_pro

Custom

Cust	Region	Type
C1	Asia	Retail
C2	Europe	Dealer
C3	Asia	Dealer
C4	America	Retail
C5	Europe	Dealer

BI on Customer.Region

v	bitmap
Asia	1 0 1 0 0
Europe	0 1 0 0 1
America	0 0 0 1 0

# Indexing OLAP Data: Bitmap Index /2

(1) Bitmap join index (BI on Fact Table Joined with Dimension tables)

- Conceptually, perform a join, map each dimension value to the ~~Assignment Project Exam Help~~ bitmap of corresponding fact table rows.

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

-- ORACLE SYNTAX –

```
CREATE BITMAP INDEX sales_cust_region_bjix
  ON sales(customer.cust_region)
  FROM sales, customer
 WHERE sales.cust_id = customers.cust_id;
```

# Indexing OLAP Data: Bitmap Index /3

Sales

time	customer	loc	Sale
101	C1	100	1
173	C1	200	2
208	C2	10	1
863	C3	20	1
991	C1	100	8
1001	C2	200	13
1966	C4	100	21
2017	C5	200	34

Customer

Cust	Region	Type
C1	Asia	Retail
D2	Europe	Dealer
	sia	Dealer
	merica	Retail
	pe	Dealer

Assignment Project Exam Help  
https://eduassistpro.github.io/  
Add WeChat edu\_assist\_pro

BI on Sales(Customer.Region)

v	bitmap
Asia	11011000
Europe	00100101
America	00000010

# Q2: Selection on high-cardinality attributes

TIME
time_key
day
day_of_the_week
month
quarter
year

Assignment Project Exam Help  
<https://eduassistpro.github.io/>  
Add WeChat `edu_assist_pro`

customer_k
location_key
<i>units_sold</i>
<i>amount</i>

CUSTOMER
customer_key
customer_name
region
type

IN

LOCATION
location_key
store
street_address
city
state
country
region

$S_{\text{city}=\text{"Kingsford"}}$

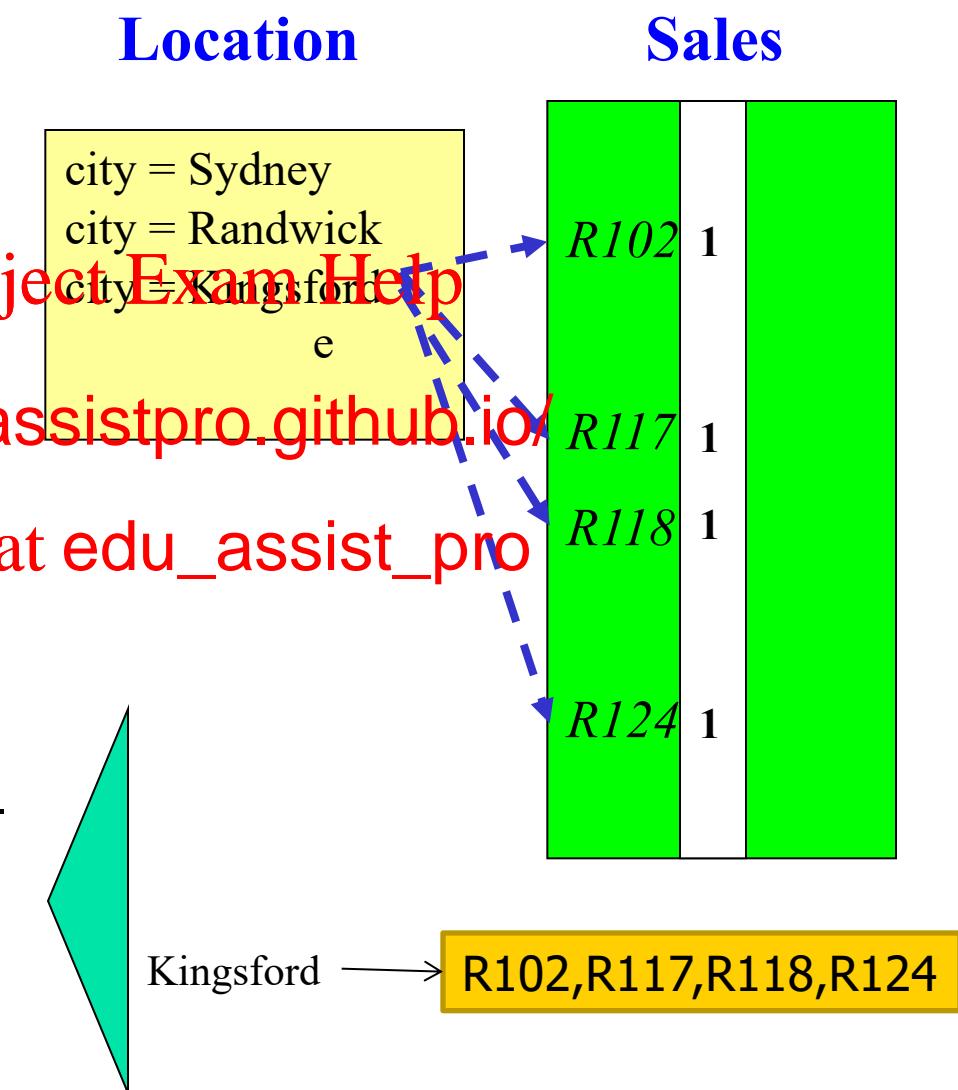
# Indexing OLAP Data: Join Indices

- Join index relates the values of the dimensions of a star schema to rows in the fact table.

- a join index on city maintains for city a list of R <https://eduassistpro.github.io/> the tuples recording the sales in the city

- Join indices can span multiple dimensions OR

- can be implemented as bitmap-indexes (per dimension)
- use bit-op for multiple-joins



# Q3: Arbitrary selections on Dimensions

TIME
time_key
day
day_of_the_week
month
quarter
year

$S_{isSchoolHolidy(day)}$

Assignment Project Exam Help

product_key
location_key
<i>units_sold</i>
<i>amount</i>

$S_{city} \sim \wedge S + ford/$

Customer
customer_key
customer_name
region
type

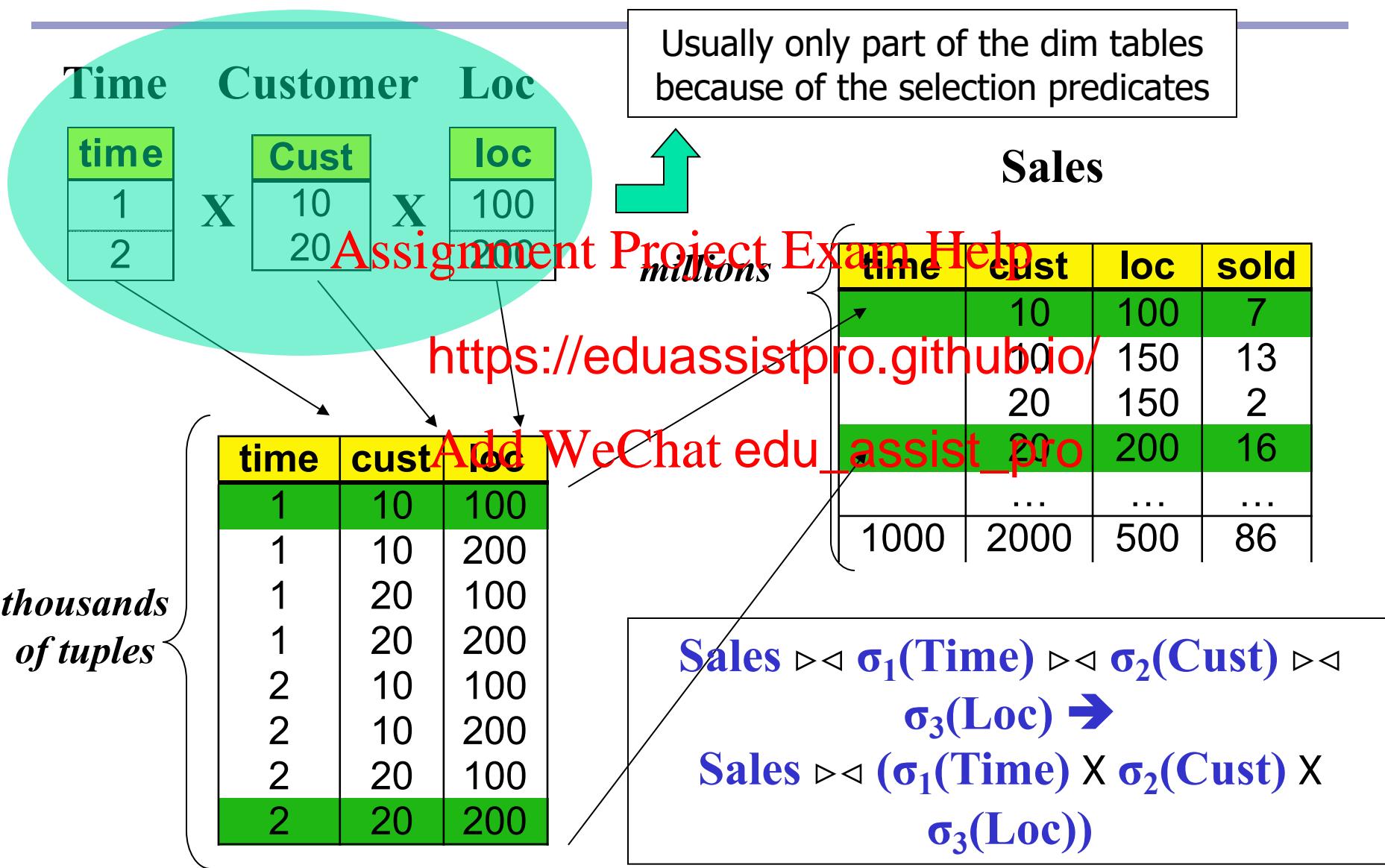
<https://eduassistpro.github.io/>

Add WeChat **edu\_assist\_pro**

IN

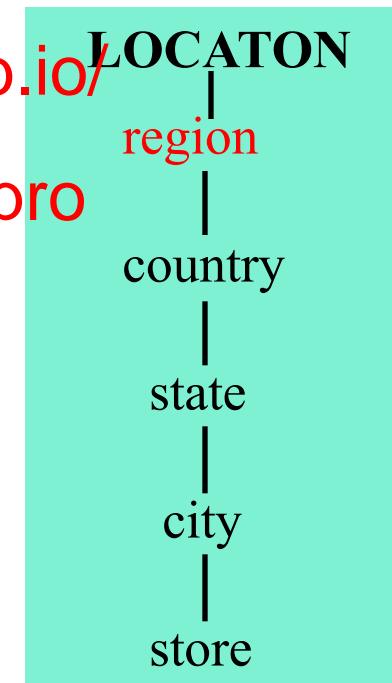
LOCATION
location_key
store
street_address
city
state
country
region

# Star Query and Star Join (Cont.)



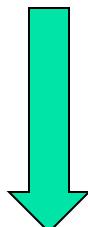
# Q4: Coarse-grain Aggregations

- “Find total sales per customer type in our stores in Europe”
  - Join-index will prune  $\frac{3}{4}$  of the data (uniform sales), but the remaining  $\frac{1}{4}$  is still large (several millions transactions)
    - Index is un <https://eduassistpro.github.io/>
- High-level aggregations are e !
  - Add WeChat edu\_assist\_pro
  - ⇒ Long Query Response Time
  - ⇒ Pre-computation is necessary
  - ⇒ Pre-computation is most beneficial



# Cuboids = GROUP BYs

- Multidimensional aggregation = selection on corresponding cuboid

$$\text{GB}_{(\text{type}, \text{city})}(\text{Assignment}(\text{Project}(\text{Exam}(\text{Help}(\text{Cuboid}(\text{Year}, \text{Type}, \text{City})))) \sigma_3(\text{Loc})) )$$


$\sigma_1$  selects some Brands,

$\sigma_3$  selects some Cities

$$\text{GB}_{(\text{type}, \text{city})}(\sigma_{1'2'3'}(\text{Cuboid}(\text{Year}, \text{Type}, \text{City})))$$

- Materialize some/all of the cuboids
  - A complex decision involving cuboid sizes, query workload, and physical organization

# Two Issues

---

- How to store the materialized cuboids?
- How to compute the cuboids efficiently?

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# CUBE BY in ROLAP

Sales		Product				
		1	2	3	4	ALL
Store	1	454	-	-	925	1379
	2	468	800	-	-	1268
	3	296	-	250	536	1082
	4	652	-	540	142	1334
	ALL	1870	800	780	1670	5120

Assignment Project Exam Help  
https://eduassistpro.github.io/

4 Group-bys here:  
**(store,product)**  
**(store)**  
**(product)**  
**0**

- Need to write 4 queries!!!
- Compute them independently

Store	Product_key	sum(amount)
1	1	454
1	4	925
2	1	468
2	2	800
3	1	296
3	2	240
4	1	625
4	3	240
ALL	4	745
ALL	ALL	1379
ALL	ALL	1268
ALL	ALL	536
ALL	ALL	1937
ALL	1	1870
ALL	2	800
ALL	3	780
ALL	4	1670
ALL	ALL	5120

**SELECT LOCATION.store, SALES.product\_key, SUM (amount)**

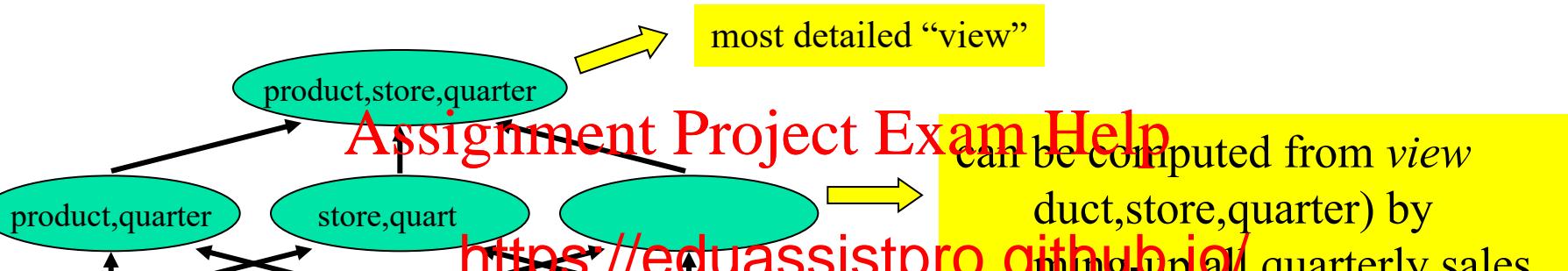
**FROM SALES, LOCATION**

**WHERE SALES.location\_key=LOCATION.location\_key**

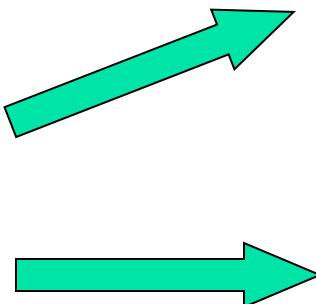
**CUBE BY SALES.product\_key, LOCATION.store**

# Top-down Approach

- Model dependencies among the aggregates:



Cuboid AB		
A	B	M
1	1	10
1	3	20
1	2	30
1	1	40
2	4	50



	B	M
1	(all)	???
2	(all)	???

Cuboid B		
A	B	M
(all)	1	???
(all)	2	???
(all)	3	???
(all)	4	???

# Bottom-Up Approach (BUC)

---

- BUC (Beyer & Ramakrishnan, SIGMOD'99)
- Ideas Assignment Project Exam Help
  - Compute th bottom up <https://eduassistpro.github.io/>
  - Divide-and-conquer
- A simpler recursive version:
  - BUC-SR

A	B	...
1	1	...
1	3	...
1	2	...
1	1	...
2	...	...



...  
■ ■ ■

# Understanding Recursion /1

---

- Powerful computing/problem-solving techniques  
**Assignment Project Exam Help**
- Examples
  - Factorial: <https://eduassistpro.github.io/>
    - $f(n) = 1$ , if  $n \leq 1$
    - $f(n) = f(n-1) * n$ , if  $n \geq 1$
  - Quick sort:
    - $\text{Sort}([x]) = [x]$
    - $\text{Sort}([x_1, \dots, \text{pivot}, \dots x_n]) = \text{sort}[ys] ++ \text{sort}[zs]$ , where

$ys = [x | x \in x_i, x \leq \text{pivot}]$

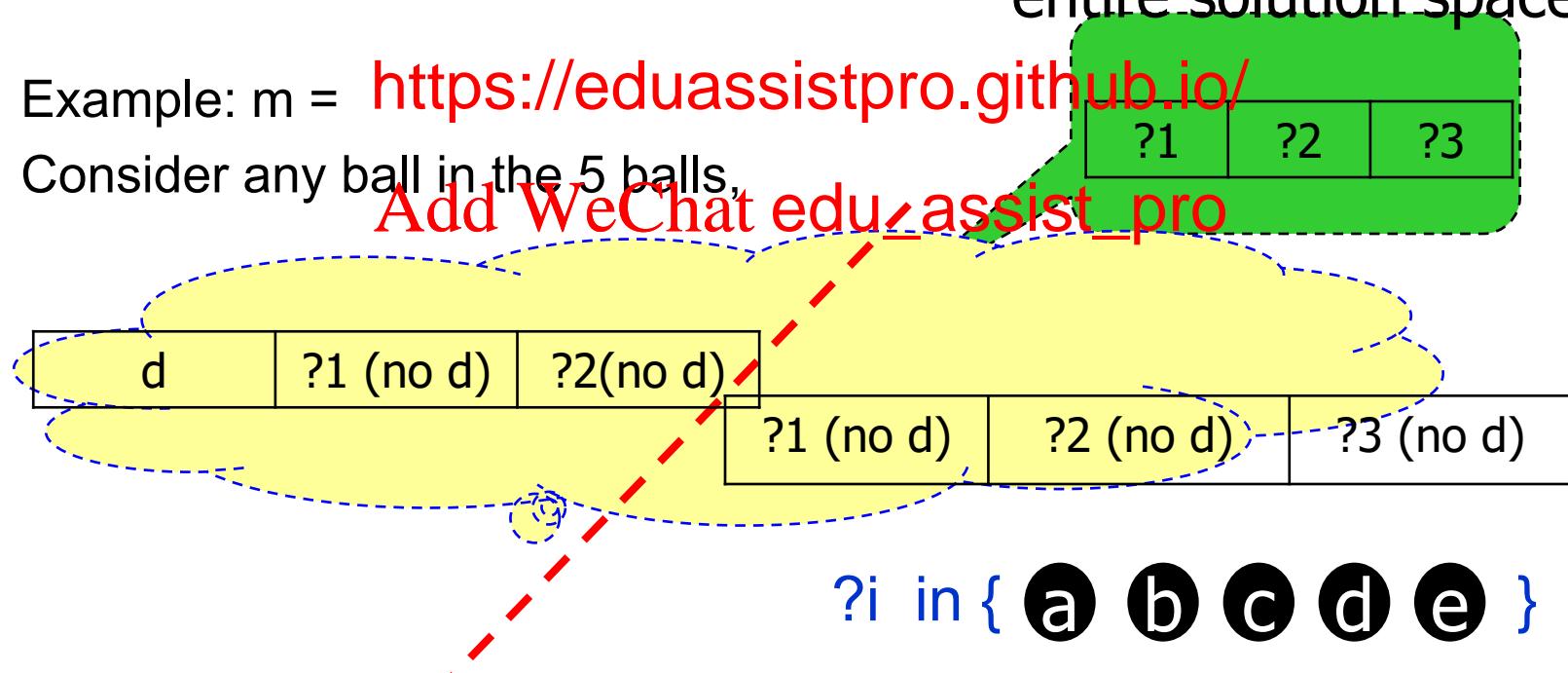
$zs = [x | x < x_i, x > \text{pivot}]$

$$f(0) = 0! = \\ ???$$

List comprehension  
in Haskell or  
python

# Understanding Recursion /2

- Let  $C(n, m)$  be the number of ways to select  $m$  balls from  $n$  numbered balls
- Show that  $C(n, m) = C(n-1, m-1) + C(n-1, m)$
- Example:  $m = \text{https://eduassistpro.github.io/}$
- Consider any ball in the 5 balls,



# Key Points

---

- Sub-problems need to be “**smaller**”, so that a simple/trivial boundary case can be reached  
*Assignment Project Exam Help*
- Divide-and
  - There ma <https://eduassistpro.github.io/> ntire solution space can be divided int sub-spaces, each of which can be co *Add WeChat **edu\_assist\_pro** cursively.*

# Geometric Intuition /1

- Reduce Cube(in 2D) to Cube(in 1D)

	b1	b2	b3	Assignment Project Exam Help
a1	M11			[Step 1]
a2	M21			[Step 1]
	[Step 2]	[Step 2]		[Step 3]

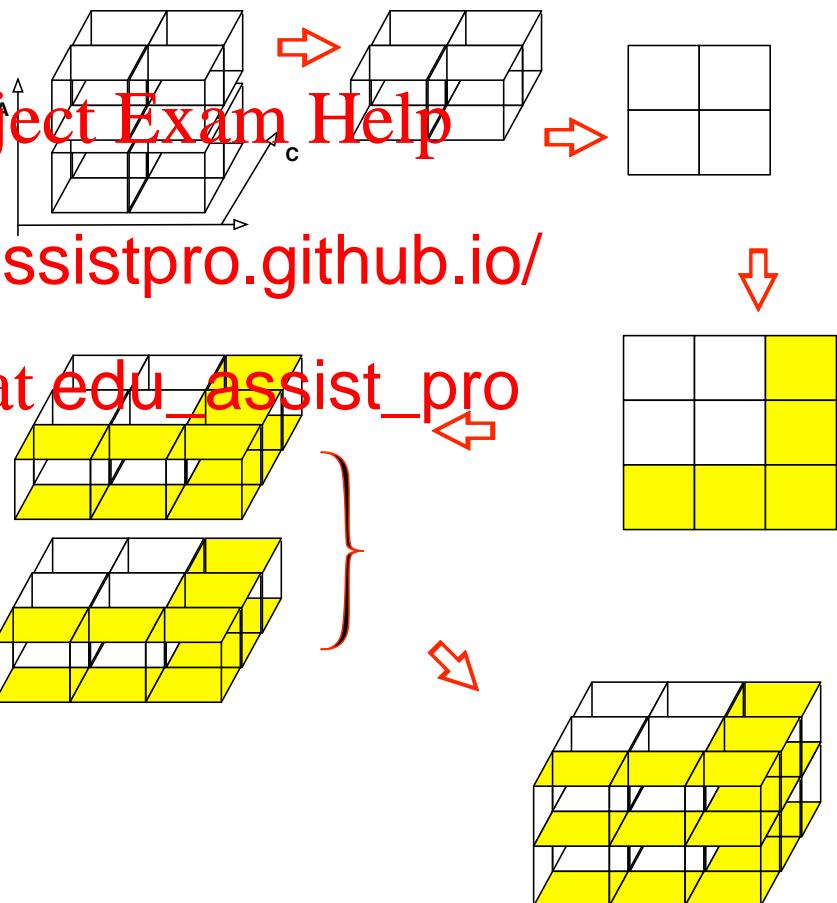
<https://eduassistpro.github.io/>  
Add WeChat edu\_assist\_pro

	b1	b2	b3	*
[a1] ×	M11	M12	M13	[Step 1]
[a2] ×	M21	M22	M23	[Step 1]
[*] ×	[Step 2]	[Step 2]	[Step 2]	[Step 3]

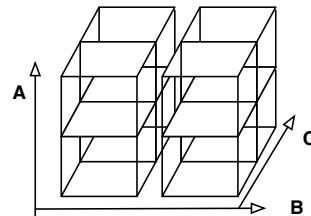
# Geometric Intuition /2

- Reduce Cube(in 3D) to Cube(in 2D)

Assignment Project Exam Help

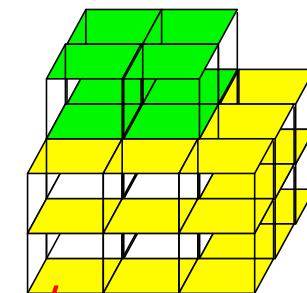
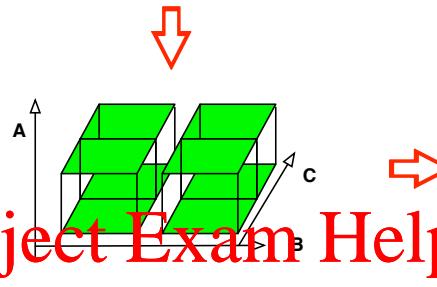


# Geometric Intuition /3



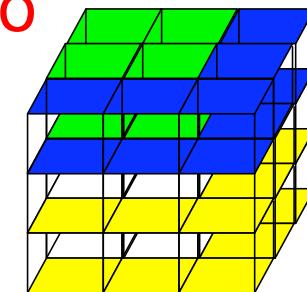
- Reduce Cube(in 3D) to Cube(in 2D)

Assignment Project Exam Help



<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# Algebraic Derivation

---

- How to compute  $n$ -dim cube on  $(n+1)$ -dim base cuboid (array)?
  - What do ~~Assignment~~ ~~output~~ ~~Project~~ ~~book~~ ~~like~~ ~~Exam~~ ~~Help~~
- How to comp <https://eduassistpro.github.io/> on  $(n+1)$ -dim base cuboid (
  - What else ~~Add WeChat~~ [edu\\_assist\\_pro](https://edu_assist_pro)

[{r1-r5}, ABC]  
[{r1-r5}, BC]

r1	A	B	C	M
r2	1	1	1	10
r3	1	1	2	20
r4	1	2	1	30
r5	1	3	1	40
	2	1	1	50

# BUC-SR (Simple Recursion)\*

- BUC-SR(data, dims)

- If (dims is empty)  
  - Output (su <https://eduassistpro.github.io/>)
- Else

- Dims = [dim1, rest\_of\_dim]

- For each distinct value  $v$  of dim1

- slice\_v = slice of data on “dim1 = v”

- BUC-SR(slice\_v, rest\_of\_dims)

- data' = Project(data, rest\_of\_dims)

- BUC-SR(data', rest\_of\_dims)

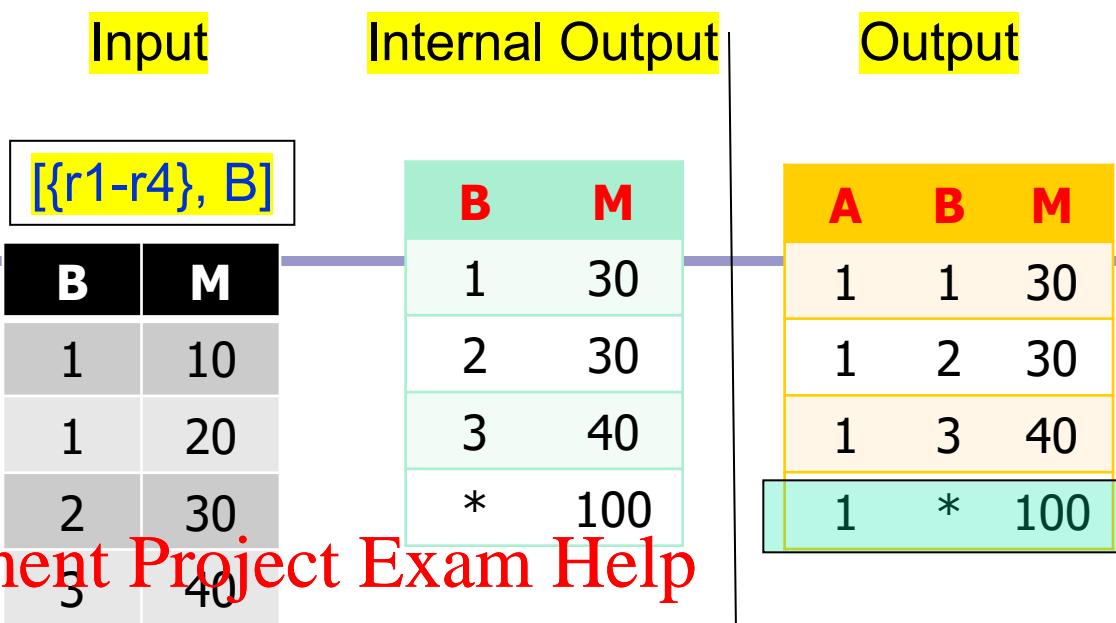
Boundary case:  
data is essentially a list  
of measure values

General case:  
1)Slice on dim1. Call  
BUC-SR recursively for  
each slice

2)Project out dim1, and  
call BUC-SR on it  
recursively

# Example

	1	2	3	*
1	30	30	40	100
2	50			50
*	80	30	40	150



Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

$\{\{r1-r5\}, AB\}$

	A	B	M
r1	1	1	10
r2	1	1	20
r3	1	2	30
r4	1	3	40
r5	2	1	50

$\{\{r1'-r5'\}, B\}$

	B	M
	1	80
	2	30
	3	40
	*	150

	B	M
	1	80
	2	30
	3	40
	*	150

	A	B	M
	2	1	50
	2	*	50

	A	B	M
	*	1	80
	*	2	30
	*	3	40
	*	*	150

# Try a 3D-Cube by Yourself

---

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

[{r1-r5}, ABC]

	A	B	C	M
r1	1	1	1	10
r2	1	1	2	20
r3	1	2	1	30
r4	1	3	1	40
r5	2	1	1	50

# MOLAP

---

- (Sparse) array-based multidimensional storage engine
- Pros: Assignment Project Exam Help
  - small size <https://eduassistpro.github.io/>
  - fast in indexing
- Cons:
  - scalability
  - conversion from relational data

# Multidimensional Array

$$f(\text{time}, \text{item}) = 4 * \text{time} + \text{item}$$

time	item	dollars_sold
Q1	home entertainment	605
Q2	home entertainment	680
Q3	home entertainment	812
Q4	home entertainment	
Q1	computer	
Q2	computer	952
Q3	computer	1023
Q4	computer	1038
Q1	phone	14
Q2	phone	31
Q3	phone	30
Q4	phone	38
Q1	security	400
Q2	security	512
Q3	security	501
Q4	security	580



Step 1

Mapping  
time      value

item	value
home entertainment	0
computer	1
phone	2
security	3

time	item	dollars_sold	offset
0	0	605	0
1	0	680	4
2	0	812	8
3	0	927	12
0	1	825	1
1	1	952	5
	1	1023	9
	1	1038	13
0	2	14	2
1	2	31	6
2	2	30	10
3	2	38	14
0	3	400	3
1	3	512	7
2	3	501	11
3	3	580	15

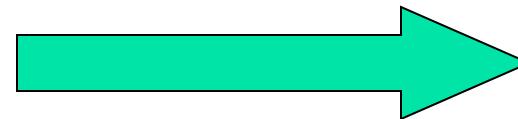
Assignment Project Exam Help  
<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# Multidimensional Array

Step 3: If **dense**, only need to store sorted slots

offset	dollars_sold
0	605
1	825
2	14
3	100
4	680
5	952
6	31
7	512
8	812
9	1023
10	30
11	501
12	927
13	1038
14	38
15	580



- Think: how to decode a slot?

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

Dense MD array
605
825
14
400
680
952
31
512
812
1023
30
501
927
1038
38
580

# The Sparse Case

$$f(\text{time}, \text{item}) = 4 * \text{time} + \text{item}$$

time	item	dollars_sold
Q1	home entertainment	605
Q4	security	580

*/\* same table but with many rows deleted to make it sparse \*/*



Step 1

Mapping

time	value
Q3	2
Q4	

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

item	value
home entertainment	0
computer	1
phone	2
security	3

time	item	dollars_sold	offset
0	0	605	0
3	3	580	15

*/\* same table but with many rows deleted to make it sparse \*/*

# Multidimensional Array

# Choice 1

offset	dollars_sold
0	605
15	580

- Think: how to decode a slot?

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro Could use further re e

- Space usage:
    - $(d+1)*n*4$  vs  $2*n*4$
  - HOLAP:
    - Store all non-base cuboid in MD array
    - Assign a value for ALL

## Choice 2