

Business Intelligence and Data Warehousing (ANL408)

- By Sabarish Nair

Recap from last week....

- Additivity in Fact Tables
- Additive Measure
- Semi-Additive Measure
- Non-Additive Measure
- Transactional Fact Table
- Periodic Fact Table
- Accumulating Snapshot
- Factless Fact Table
- Natural vs Surrogate Key
- Practical: Launch Postgres SQL
- Practical: Create a New Database
- Practical: Create a Staging Schema



Dimension Tables

- Descriptive Attributes
- Used to group and filter (slice and dice)
- Always has a primary key (can be surrogate)
- Related to fact table using a primary key-foreign key relationship

Date Dimensions

- Commonly Used Dimension
- Contains date related features (year, month, month-name, etc.)
- Surrogate Key is usually the date itself (not incremental IDs)
 - E.g. 2024-09-03 --> 20240903
- Handle NULL values (set default date --> 1900-01-01)
- Date may have time stamp
- Timestamp is usually a separate dimension



Date Features

- Can be a combination of numbers and text (01, January, etc.)
- Can be a combination of attributes (Q1-2024)
- Can be long or abbreviated (Jan, Mon)
- Can have flags (Company Holidays, Weekend, etc.)

Date_PK	Year	Month	Day
20240309	2024	03	09
20240307	2024	03	07



Nulls in Dimensions

- Recap
 - Avoid NULLs in Foreign Keys (Replace with dummy values)
 - NULLs break referential integrity as they don't appear in JOINS
 - NULLs can be a part of fact table (e.g. no sales on weekends)
- Replace NULLS with descriptive values (e.g. "Unkown", "No description available")
- More understandable for business users
- Aggregated values appear in Business Tools

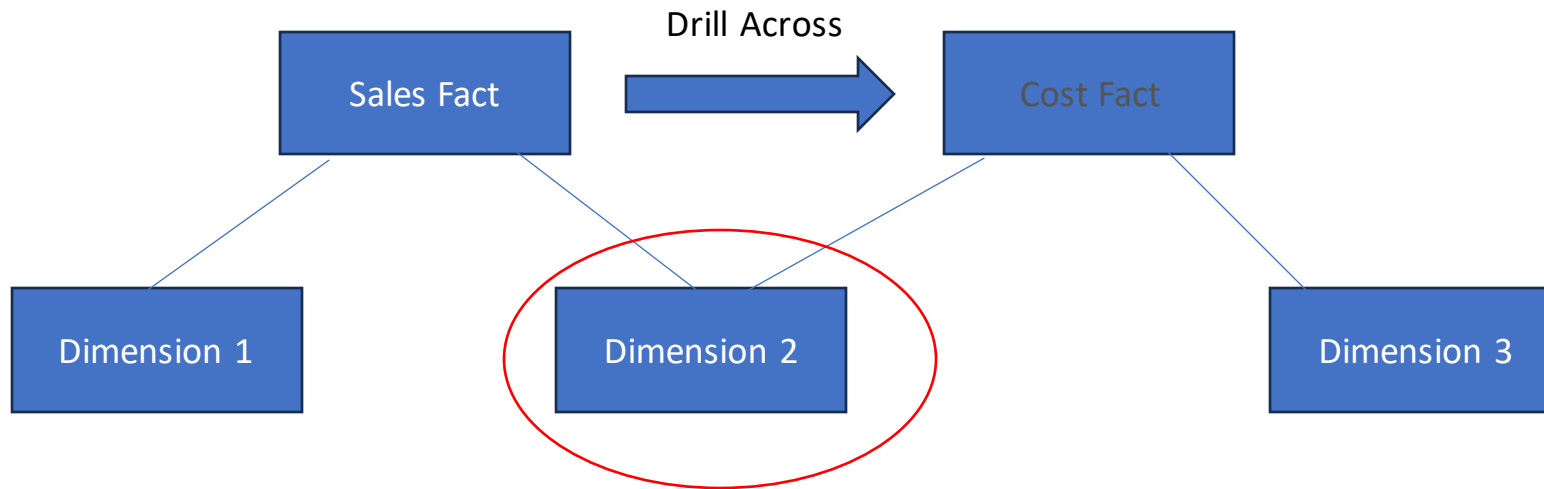
Hierarchies in Dimensions

- Data is often normalized.
- Normalized data suitable for transactional purposes (Write-heavy).
- DW is Read-Heavy (Used for analytical purposes)
- Snowflake Schema should be avoided (Multiple Hierarchies)
- Try to avoid normalization as much as possible (flatten the table)



Conformed Dimensions

Dimensions shared by multiple fact tables/stars
For example: Region, Date, etc.





Degenerate Dimensions

- Dimensions that are derived from fact table columns rather than being stored in separate dimension tables.
- They do not have their own dedicated dimension tables.
- No additional attributes or relationship.
- Mainly used for reporting and analysis purposes
- For example: Transaction number, order number or any other unique identifier, etc.

Example: Degenerate Dimension

Transaction_PK	Amount	Payment_FK
1	200	203-111
2	300	203-112
3	400	204-116

Transactional Fact Table



Junk Dimension

Dimensions with various flags/indicators with low cardinality (not many value, e.g. Yes/No, etc.)

E.g. A box where we store items, but no separate storing location.

Terminology is used internally. To business users its "transactional indicator dimension".

Simplifies reporting and analysis.

Avoids cluttering of main dimension tables.

Junk Dimension: Example

Transaction_ PK	Amount	Payment_Type	Incoming/Outbound	Is Bonus
1	200	Credit Card	Incoming	Yes
2	200	Debit Card	Outbound	No
3	400	Cash	Incoming	Yes



Transaction_ PK	Amount	Payment_Flag_FK
1	200	1
2	200	2
3	400	3

Flag_ PK	Payment_T ype	Incoming/ Outbound	Is Bonus
1	Credit Card	Incoming	Yes
2	Debit Card	Outbound	No
3	Cash	Incoming	Yes

Role Playing Dimension



- Referenced multiple times by Fact Table
- Dimensions that serve multiple roles within single fact table.
- Represents different perspectives or viewpoints of data.
- Each instance of the dimension table in the fact table is referred to as a "role."
- Example: Order Date, Ship Date, Delivery Date

Example: Role Playing Dimension

Order_id	Order_date_FK	Production_Date_FK
1	20240901	20240804
2	20240902	20240901



Date_PK	Year	Month	Day
20240309	2024	03	09
20240307	2024	03	07

Practical



Please download the ProductsData.csv file from Moodle (present under week 6)



Please have Postgres installed in your laptops.



Create a new database named "DataWarehouse".



Create a new schema named "Staging" within the database.

Create the staging table

- Create a new table using the table creation wizard.
- Right click on the tables section.
- Fill in the required details.

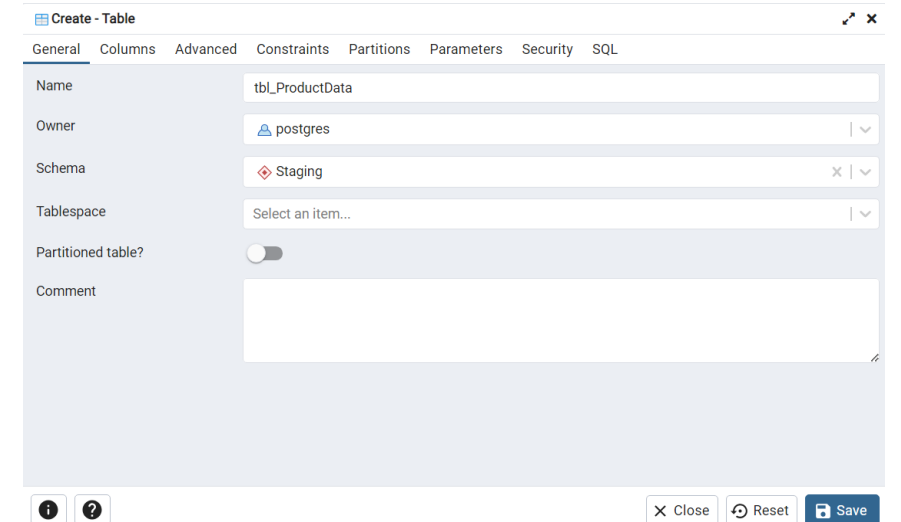
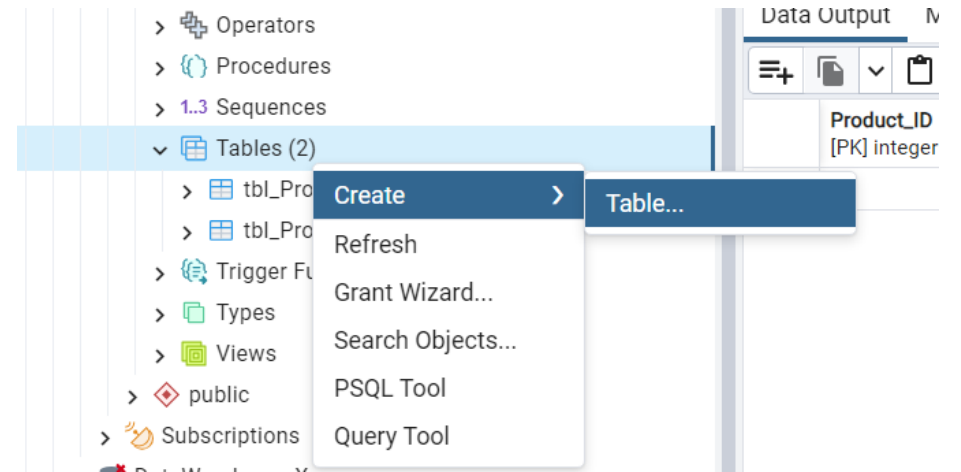
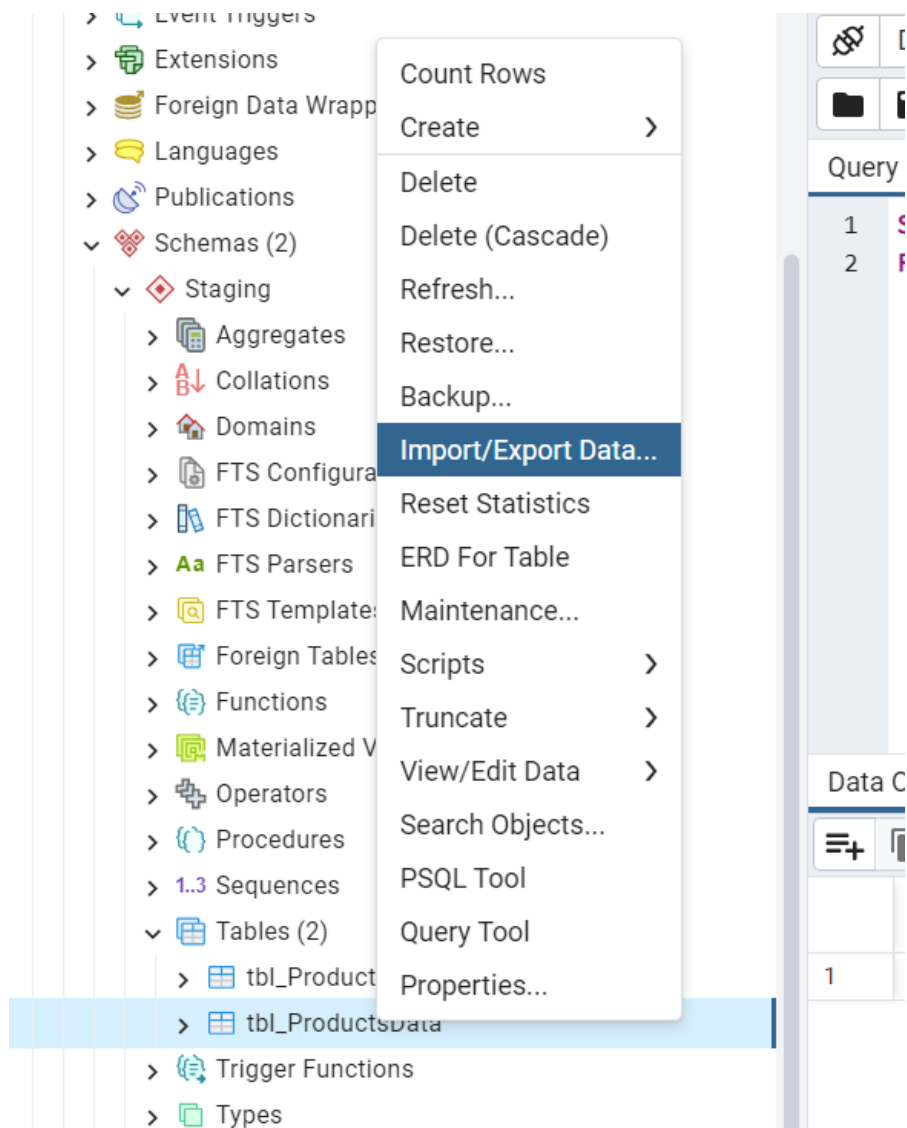


Table Creation Script

```
CREATE TABLE "Staging"."tbl_ProductsData"  
(  
    sales_id integer,  
    date_sales date,  
    product_id integer,  
    product_name character varying,  
    category character varying,  
    price numeric,  
    customer_id integer,  
    customer_name character varying,  
    city character varying,  
    country character varying,  
    PRIMARY KEY (sales_id)  
);
```





Import the CSV data to the staging table

- Right click on the table name and select the option "Import/Export Data."

Import/Export data - table 'tbl_ProductsData'

General Options Columns

Import/Export ☒ Import ☐ Export

Filename E:\Companies\PMG\Ireland2ndInnings\Tutor\CityCollegeDublin\Presenta

Format csv

Encoding Select an item...

Close Reset OK

Import/Export data - table 'tbl_ProductsData'

General Options Columns

OID ☐

Header ☒

Delimiter ,

Quote "

Escape .


Close Reset OK

Select the required options

- File path, Header, Delimiter, Etc.


Process completed

Copying table data 'Staging.tbl_ProductsData' on database 'DataWarehouse2' and server 'PostgreSQL 16 (localhost:5432)'

 View Processes

Process started

Copying table data 'Staging.tbl_ProductsData' on database 'DataWarehouse2' and server 'PostgreSQL 16 (localhost:5432)'

 View Processes

Query

Query History

Scratch Pad

1

SELECT

sales_id, date_sales, product_id, product_name, category, price,

customer_id, customer_name, city, country

FROM "Staging"."tbl_ProductsData";

Data Output

Messages

Notifications

+

+

+

+

+

+

+

+

sales_id

[PK] integer

date_sales

date

product_id

integer

product_name

character varying

category

character varying

price

numeric

customer_id

integer

customer_name

character varying

city

character varying

1

1

2022-01-01

1

Laptop

Electronics

1200

1

John Doe

New York

2

2

2022-01-02

2

Smartphone

Electronics

800

2

Jane Smith

London

3

3

2022-01-03

3

Speaker

Electronics

[null]

3

[null]

Paris

4

4

2022-01-04

4

TV

Electronics

1500

[null]

[null]

Berlin

5

5

2022-01-05

5

Tablet

Electronics

-100

5

Michael Lee

Tokyo

6

6

2022-01-06

[null]

Mouse

Electronics

25

6

[null]

Sydney

Verify the data

A photograph of a desk setup. In the background, a portion of a laptop keyboard is visible, showing keys for function keys (F3-F10) and alphanumeric keys (6-0, P, L, ;, >, ?). In the foreground, a brown paper envelope is partially open. A white rectangular card is placed on top of the envelope, featuring the words "Thank you" written in a black, elegant cursive script. A black ballpoint pen with a silver-colored clip and tip lies diagonally across the bottom left of the white card and the envelope. The entire scene is set against a light-colored, horizontally-grained wooden surface.

Thank you