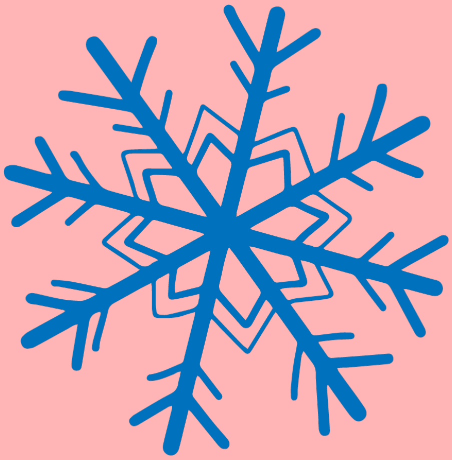
NOTES:

Star schema

Vs

Snowflake schema

Choose the right of the above



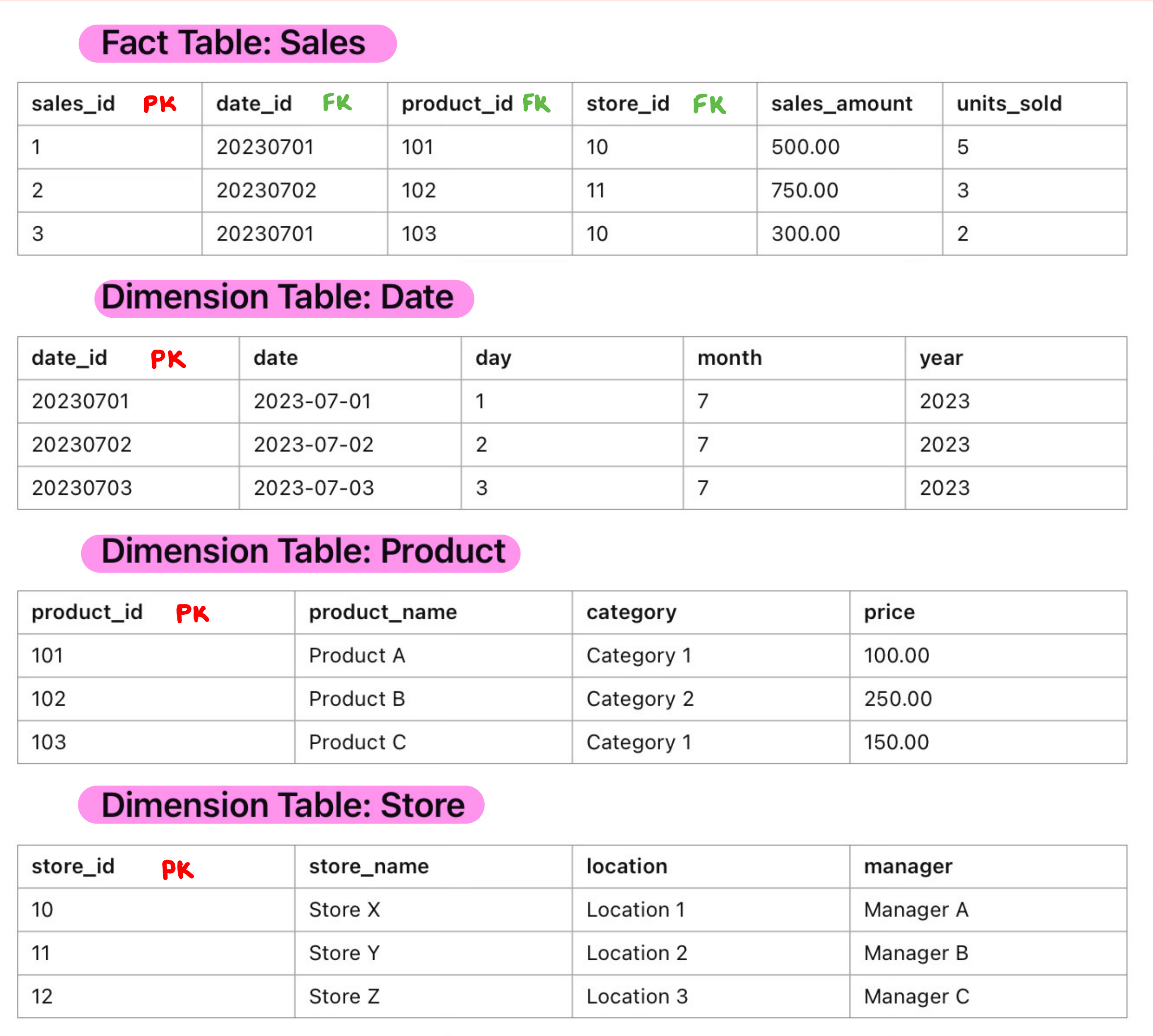
Follow me on
https://www.linkedin.com/in/dhwanimehta05/

In data warehousing, two common schemas used to organize data are the star schema and the snowflake schema. Let's explore these schemas, their use cases, and their differences with examples.

Star Schema

Definition:
The star schema is a type of database schema that organizes data into fact and dimension tables. The fact table is centralized and contains the keys to dimension tables, which are denormalized. It is called "star schema" because the diagram of the schema resembles a star⭐, with the fact table at the center and the dimension table radiating outward.

Example:
Imagine a retail company that wants to analyze sales data.



Fact Table: Sales

sales_id PK	date_id FK	product_id FK	store_id FK	sales_amount	units_sold
1	20230701	101	10	500.00	5
2	20230702	102	11	750.00	3
3	20230701	103	10	300.00	2

Dimension Table: Date

date_id PK	date	day	month	year
20230701	2023-07-01	1	7	2023
20230702	2023-07-02	2	7	2023
20230703	2023-07-03	3	7	2023

Dimension Table: Product

product_id PK	product_name	category	price
101	Product A	Category 1	100.00
102	Product B	Category 2	250.00
103	Product C	Category 1	150.00

Dimension Table: Store

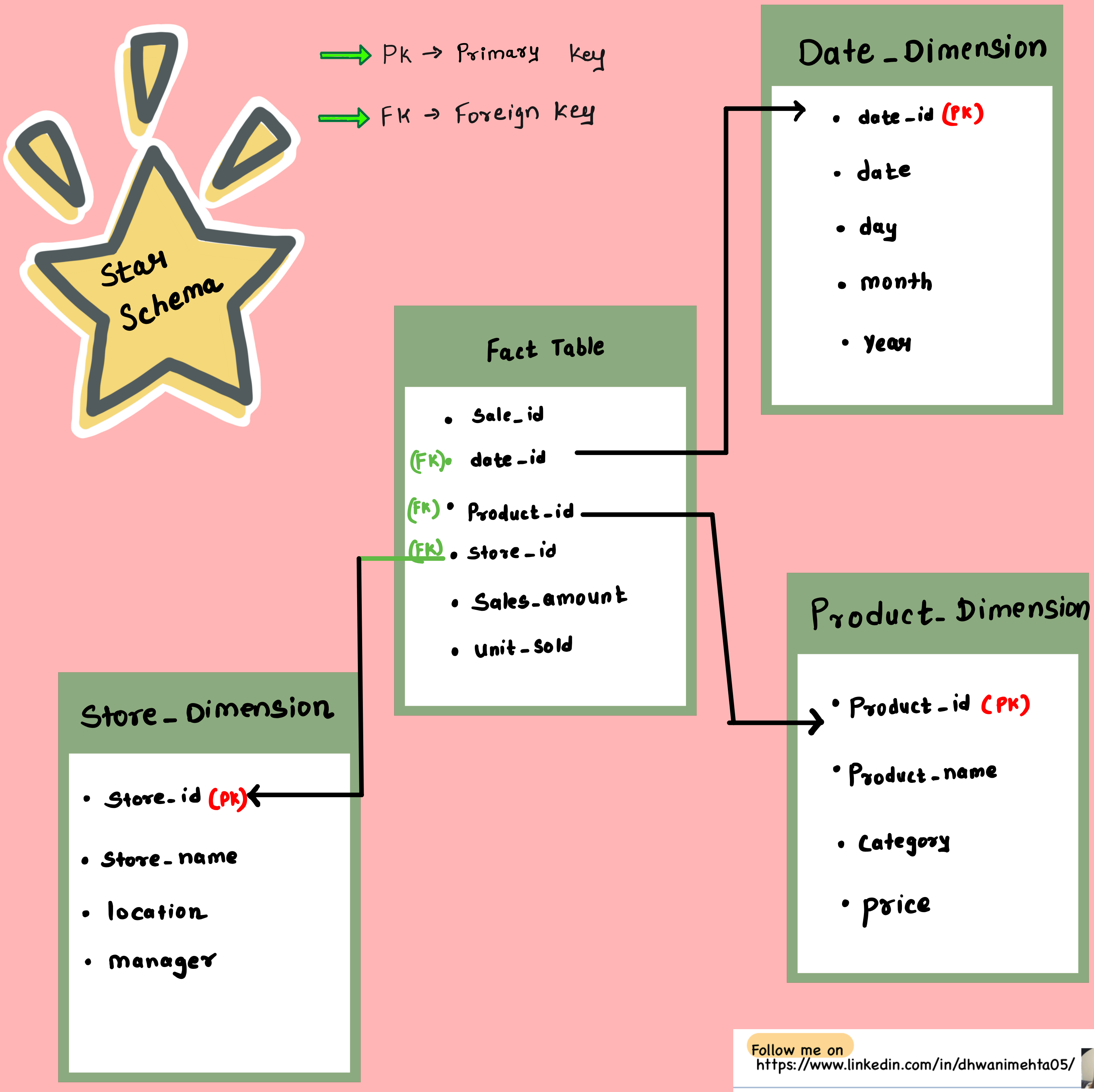
store_id PK	store_name	location	manager
10	Store X	Location 1	Manager A
11	Store Y	Location 2	Manager B
12	Store Z	Location 3	Manager C

Business Use Case:

A business can use a star schema to quickly generate sales reports, analyze sales performance over time, and track product performance across different stores.

When to Use Star Schema:

- When you need a simple, easy-to-understand schema.
- When query performance is critical, as the denormalized structure reduces the number of joins required.
- When the data is relatively stable and not subject to frequent changes.



PK → Primary key

FK → Foreign key

Fact Table

- Sale_id
- (FK) date_id
- (FK) Product_id
- (FK) store_id
- Sales_amount
- unit_sold

Date_Dimension

- date_id (PK)
- date
- day
- month
- year

Product_Dimension

- Product_id (PK)
- Product_name
- Category
- price

Store_Dimension

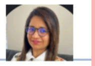
- Store_id (PK)
- Store_name
- location
- manager

In this star schema:

1. Fact Table (Sales): Contains the transactional data (e.g., sales amounts and units sold) along with foreign keys referencing the dimension tables.
2. Dimension Tables (Date, Product, Store): Provide descriptive attributes related to time, product, and store, respectively.

This design is optimal for quick and efficient querying because all descriptive attributes are immediately accessible without needing to join additional tables. The star schema is especially useful for generating reports and performing data analysis, allowing businesses to gain insights quickly.

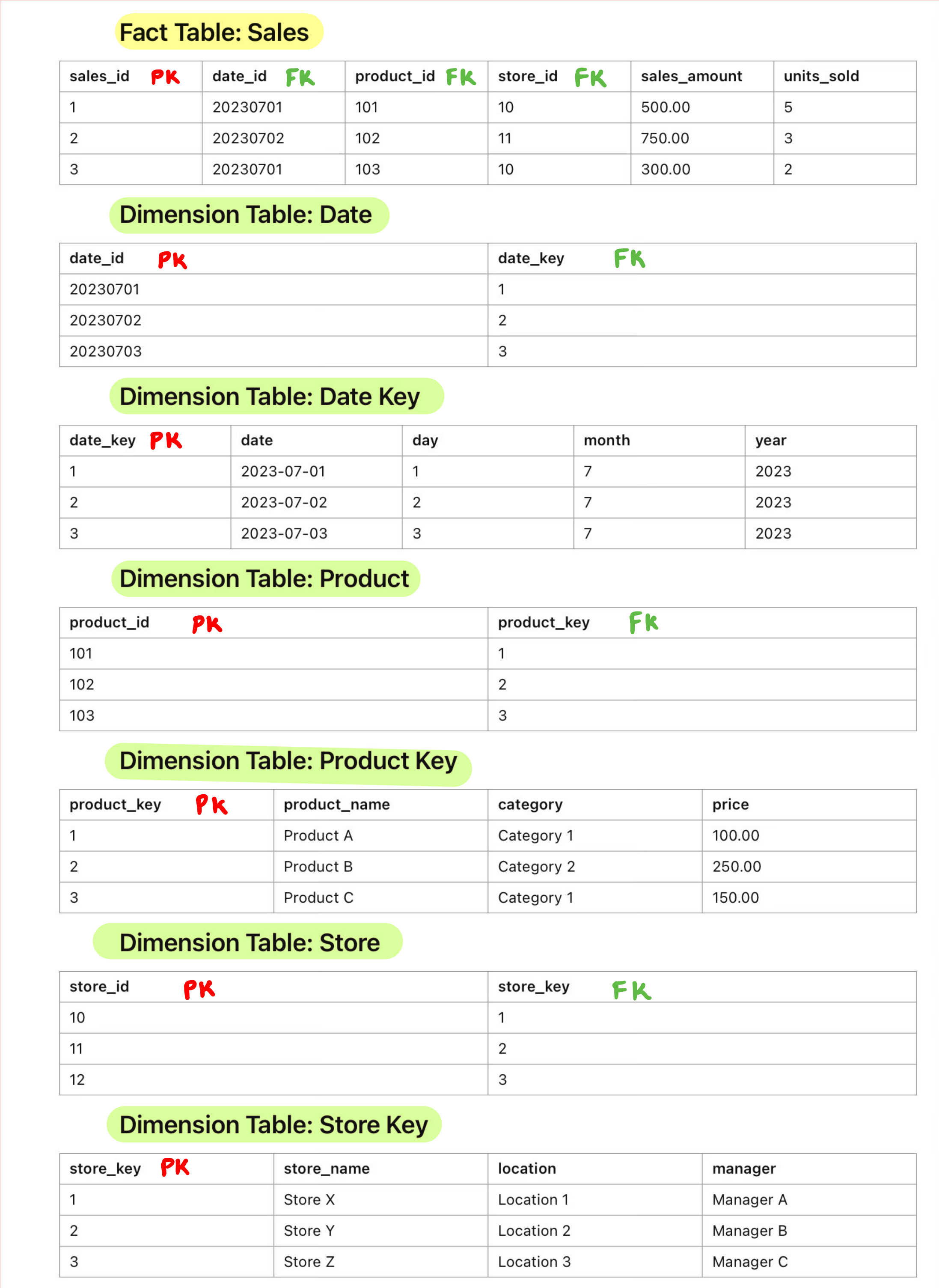
The tables are straightforward and denormalized, making it easier for analysts to understand and use the data for various business intelligence purposes.



Snowflake Schema

Definition:
The snowflake schema is a more complex schema that normalizes the dimension tables into multiple related tables, creating a "snowflake" shape. This reduces data redundancy and improves data integrity.

Example:
Using the same retail company example, the snowflake schema would further



Fact Table: Sales

sales_id PK	date_id FK	product_id FK	store_id FK	sales_amount	units_sold
1	20230701	101	10	500.00	5
2	20230702	102	11	750.00	3
3	20230701	103	10	300.00	2

Dimension Table: Date

date_id PK	date_key FK
20230701	1
20230702	2
20230703	3

Dimension Table: Date Key

date_key PK	date	day	month	year
1	2023-07-01	1	7	2023
2	2023-07-02	2	7	2023
3	2023-07-03	3	7	2023

Dimension Table: Product

product_id PK	product_key FK
101	1
102	2
103	3

Dimension Table: Product Key

product_key PK	product_name	category	price
1	Product A	Category 1	100.00
2	Product B	Category 2	250.00
3	Product C	Category 1	150.00

Dimension Table: Store

store_id PK	store_key FK
10	1
11	2
12	3

Dimension Table: Store Key

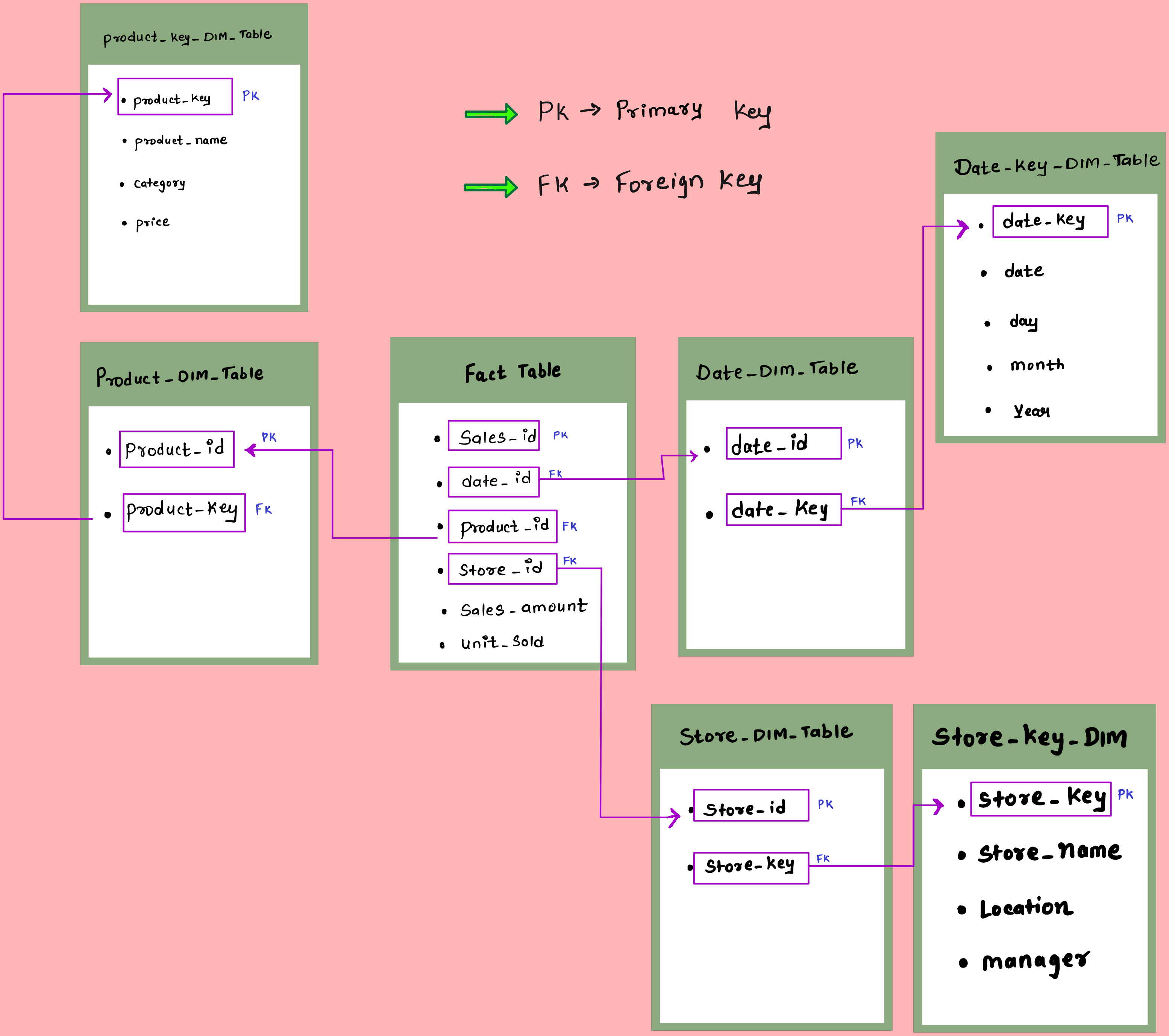
store_key PK	store_name	location	manager
1	Store X	Location 1	Manager A
2	Store Y	Location 2	Manager B
3	Store Z	Location 3	Manager C

Business Use Case:

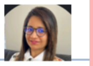
A business can use a snowflake schema to maintain data integrity, ensure that changes in product categories or store locations are consistently reflected, and minimize storage requirements.

When to Use Snowflake Schema:

- When data integrity and storage efficiency are more important than query performance.
- When the data model is complex and requires normalization to handle many-to-many relationships.
- When the data is frequently updated or changed.



Follow me on
https://www.linkedin.com/in/dhwanimehta05/



In this snowflake schema:

1. Fact Table (Sales): Contains the transactional data with foreign keys referencing the dimension tables.
2. Dimension Tables (Date, Product, Store): Provide primary keys that reference additional tables to get descriptive attributes.
3. Additional Dimension Tables (Date Key, Product Key, Store Key): Store the descriptive attributes, reducing redundancy by normalizing the data.

This design improves data integrity and reduces redundancy, ensuring that changes in descriptive data (like product categories or store locations) are consistent across the database. However, it can be more complex and may require additional joins in queries, potentially impacting performance.

The snowflake schema is useful when the data model is complex, with many-to-many relationships, and when maintaining data integrity and reducing redundancy are priorities. It allows businesses to ensure consistent and accurate data across all dimensions.

In easy language :

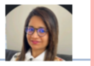
Think of the star schema as a simple, efficient shopping mall layout where you can quickly find everything you need because everything is nearby. It's great for quick shopping trips (queries), but you might notice the same items (data) repeated in multiple places.

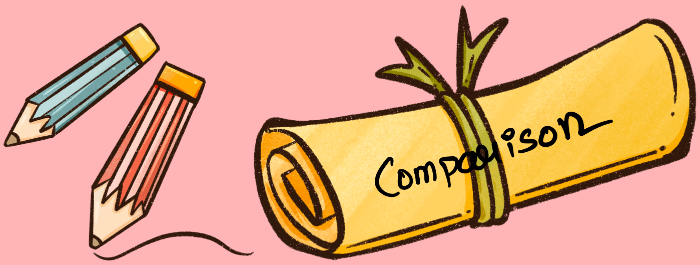
On the other hand, the snowflake schema is like a well-organized library. It takes a bit longer to find what you're looking for because books (data) are organized into many detailed sections (tables), but everything is neatly placed with no duplicates, ensuring everything is accurate and up-to-date.

Choosing between these schemas depends on your business needs:

- If you need speed and simplicity for reporting and analysis, go with the star schema.
- If you need to maintain data integrity and reduce storage costs, especially in a dynamic data environment, the snowflake schema is the better choice.

By understanding these schemas, businesses can design their data warehouses to meet their specific needs, balancing performance, integrity, and complexity.





Comparison

Aspect	Star Schema	Snowflake schema
Structure	Simple, denormalized	Complex, normalized
Performance	Faster query performance	Slower, query performance due to joins
Data Redundancy	Higher due to denormalization	Lower due to normalization
Ease of use	Easier to Understand and use	More complex to design and Query
Data Integrity	Lower potential for data anomalies	Higher, ensures Consistency
Storage	Use more storage space	More efficient storage usage

Follow me on
https://www.linkedin.com/in/dhwanimehta05/