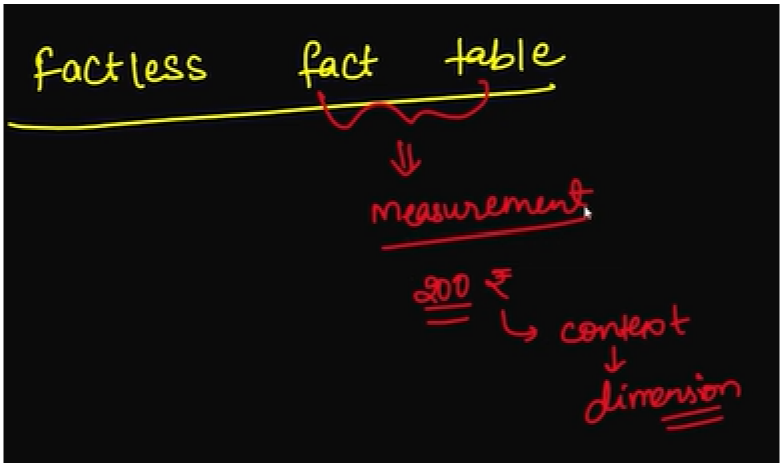
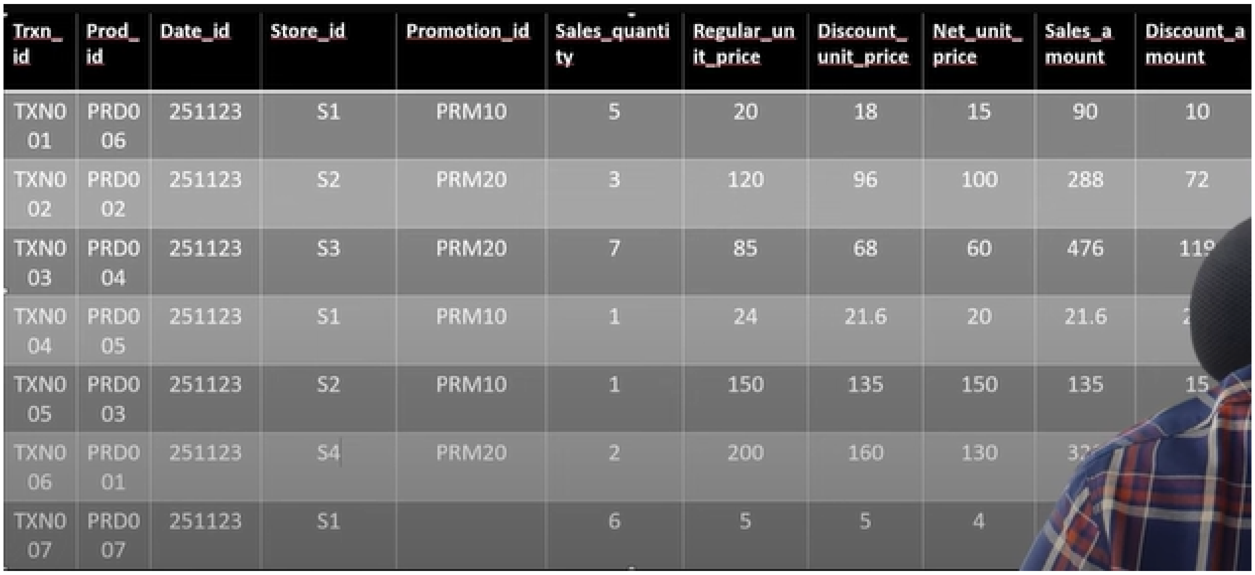
Factless Fact Table

1. Here we know fact table tells us about the measurement ..and to get the context of this data..we need dimension table

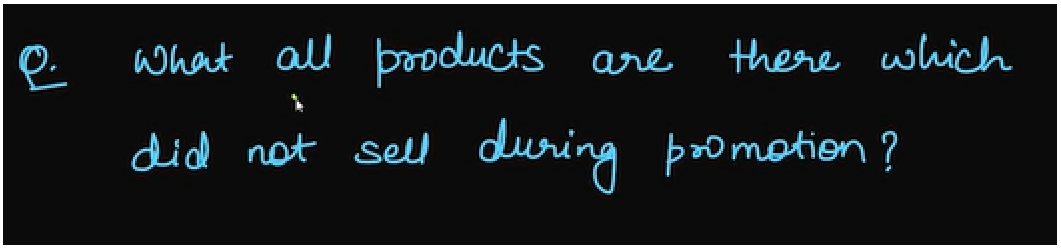


2. Here in the factless table entire row is a fact
3. Consider these transaction fact table as an example



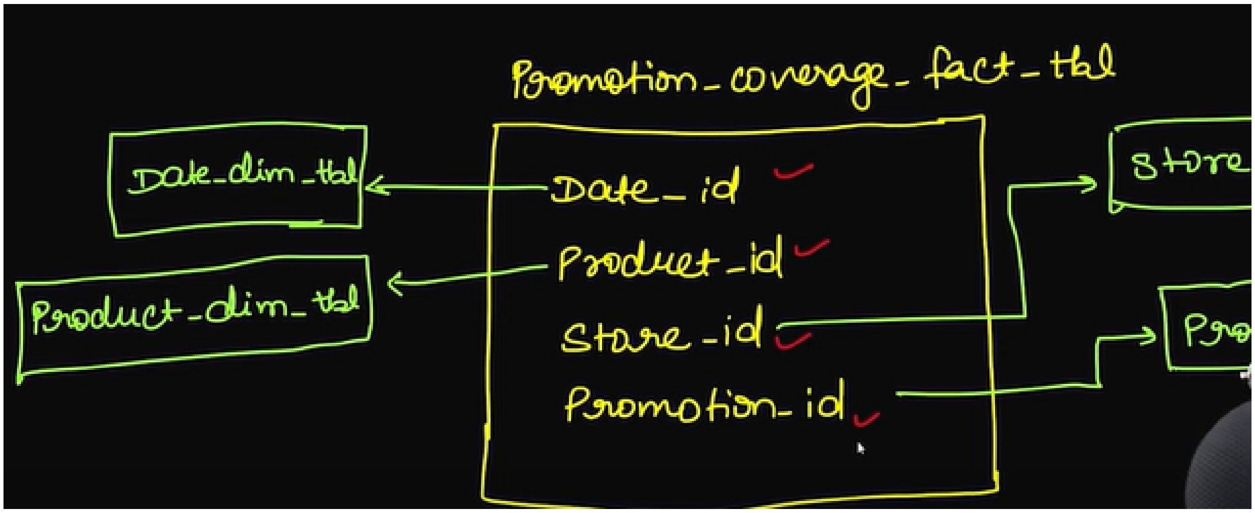
Trxn_ id	Prod_ id	Date_id	Store_id	Promotion_id	Sales_quantity	Regular_unit_price	Discount_unit_price	Net_unit_price	Sales_amount	Discount_amount
TXN001	PRD006	251123	S1	PRM10	5	20	18	15	90	10
TXN002	PRD002	251123	S2	PRM20	3	120	96	100	288	72
TXN003	PRD004	251123	S3	PRM20	7	85	68	60	476	119
TXN004	PRD005	251123	S1	PRM10	1	24	21.6	20	21.6	2
TXN005	PRD003	251123	S2	PRM10	1	150	135	150	135	15
TXN006	PRD001	251123	S4	PRM20	2	200	160	130	32	[illegible]
TXN007	PRD007	251123	S1		6	5	5	4	[illegible]	[illegible]

4. Consider these question



Q. What all products are there which did not sell during promotion?

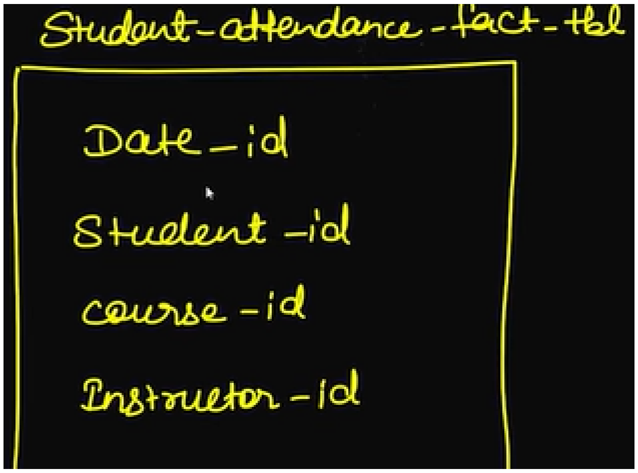
5. To answer these…we'll create a sample promotion coverage fact table



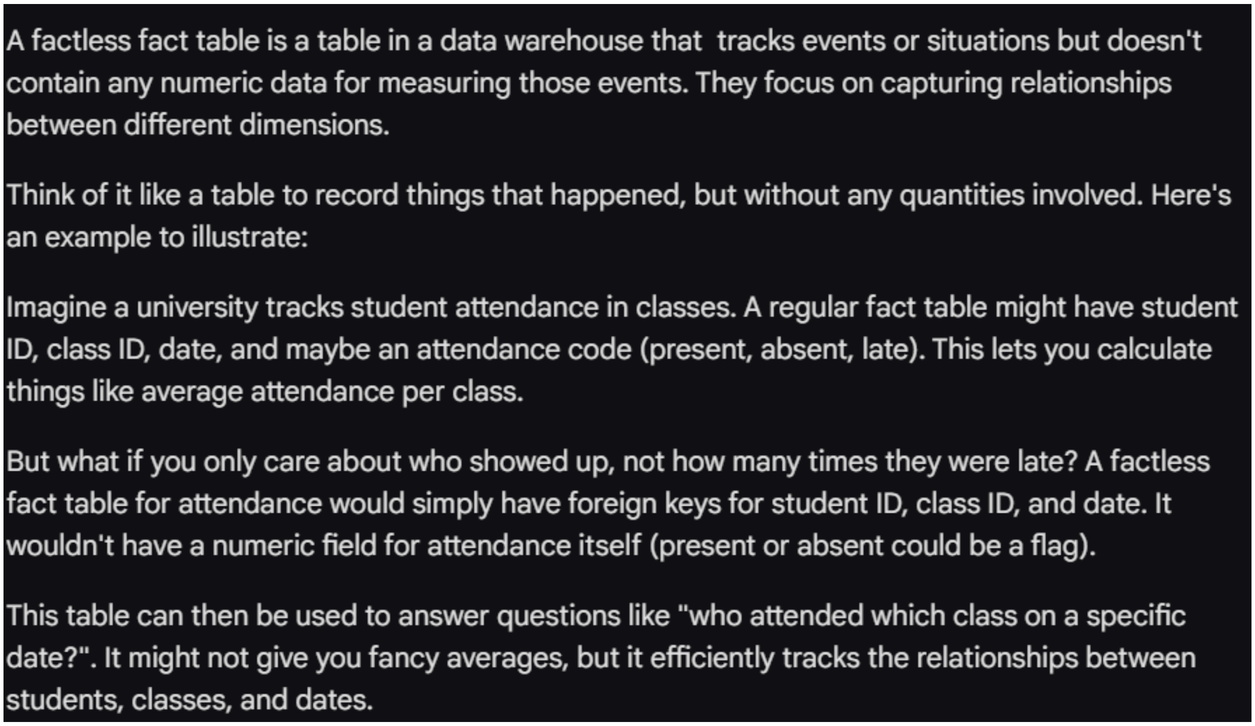
6. Here in this data ..it tells us on which date and which store the product has the promotional code…and if we observe..it does not have any measurement column
7. So now to answer our question …we have to join our two tables
8. Lets consider example 2

ex.2 Attendance (75% Attendance)

9. here in our clg we need to have 75% of attendance



10. Lets build this sample table

11. 

A factless fact table is a table in a data warehouse that tracks events or situations but doesn't contain any numeric data for measuring those events. They focus on capturing relationships between different dimensions.

Think of it like a table to record things that happened, but without any quantities involved. Here's an example to illustrate:

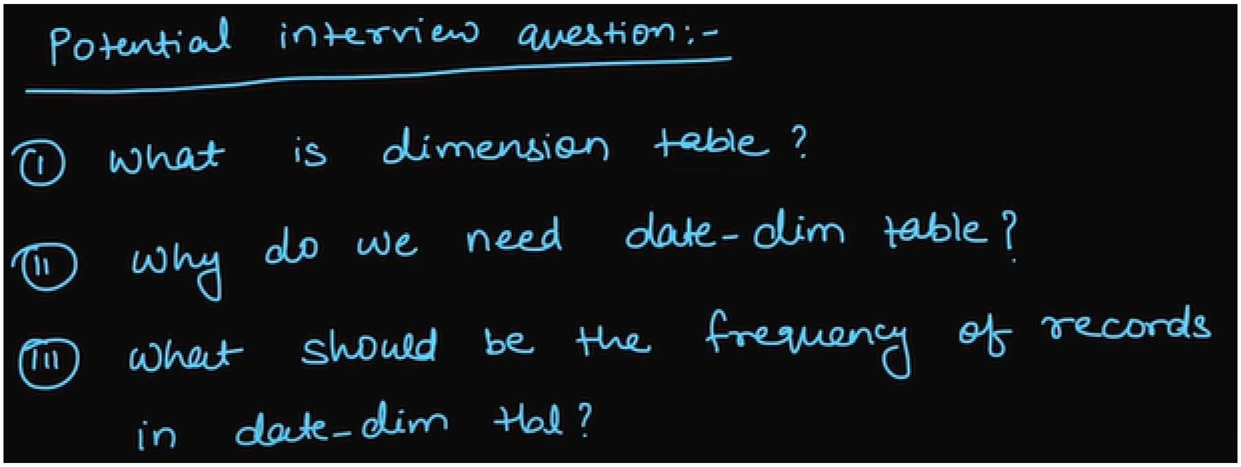
Imagine a university tracks student attendance in classes. A regular fact table might have student ID, class ID, date, and maybe an attendance code (present, absent, late). This lets you calculate things like average attendance per class.

But what if you only care about who showed up, not how many times they were late? A factless fact table for attendance would simply have foreign keys for student ID, class ID, and date. It wouldn't have a numeric field for attendance itself (present or absent could be a flag).

This table can then be used to answer questions like "who attended which class on a specific date?". It might not give you fancy averages, but it efficiently tracks the relationships between students, classes, and dates.

Date Dimension Table

1. Potential interview questions



Potential interview question:-

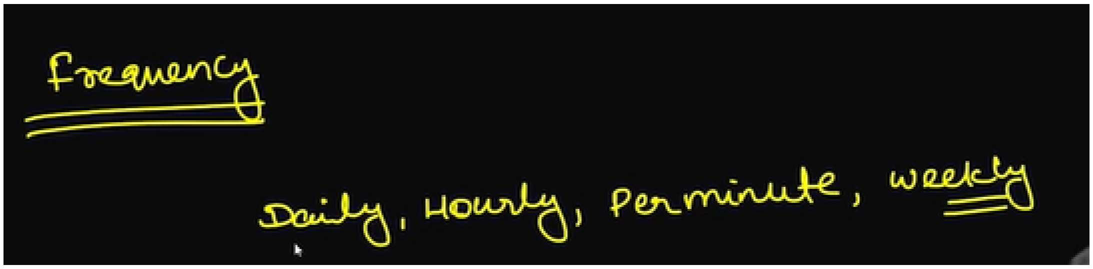
(i) what is dimension table ?

(ii) why do we need date-dim table?

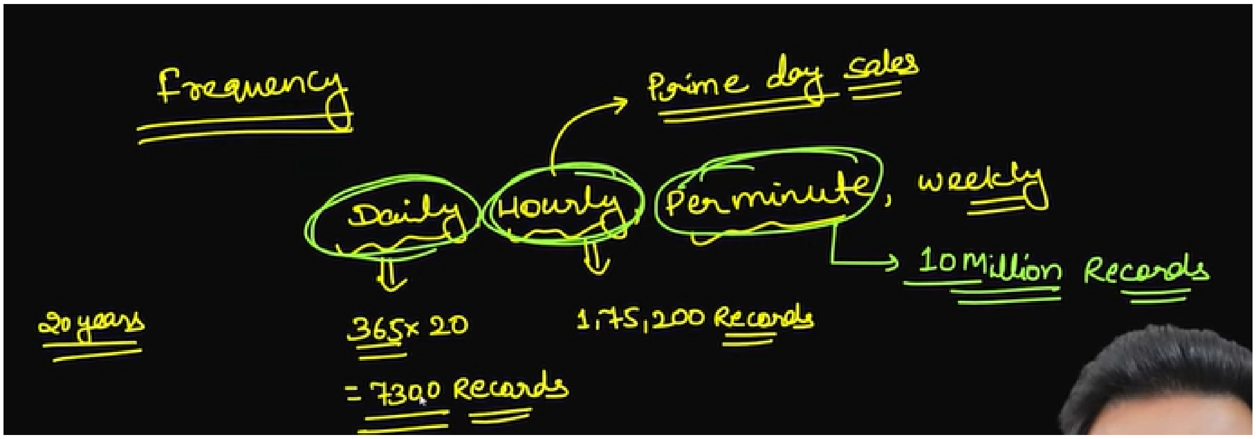
(iii) what should be the frequency of records in date-dim tbl ?

2. We'll focus on date dim table
3. What is dim table? A table which gives context to our data it is called as dim table
4. Date Dim Table

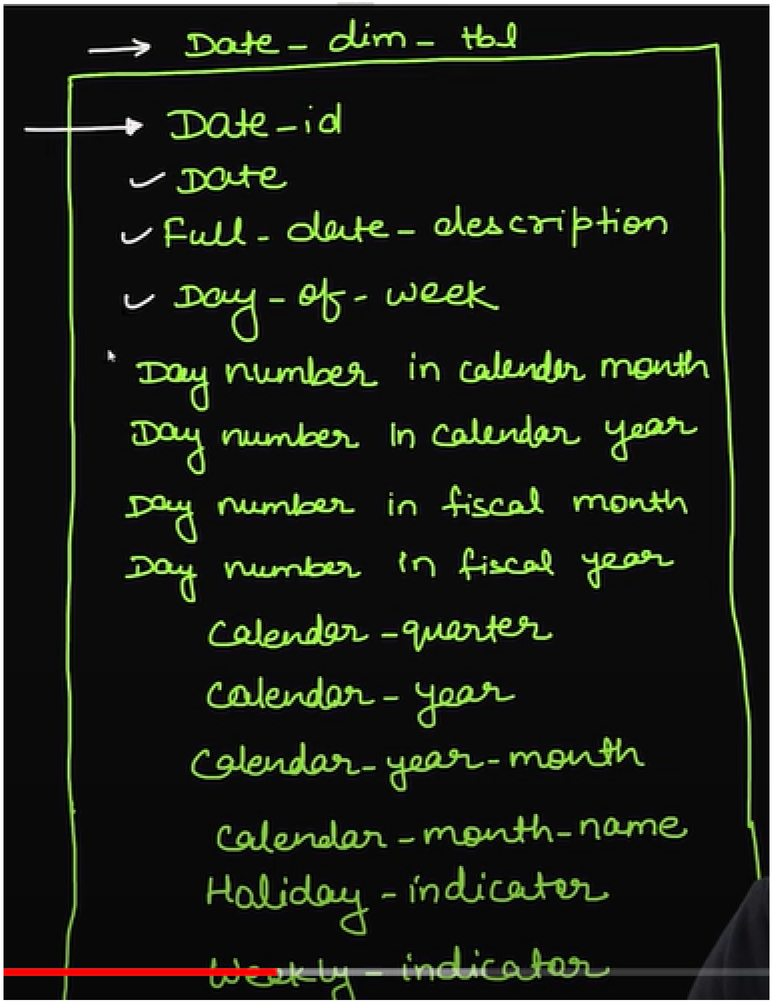
5. Here we can build our data on frequency



6. Like in each row ..we can store daily data,hourly data,,etc
7. Complications if we choose diff diff freq



8.
9. Here we want to store the data for 20yrs….if we store daily..we get 7,300 records in table..and if we store hourly we get 1,75,200 records
10. So in real time..we'll have daily dim table most of the time….and 1% hourly date dim table
11. Lets build data dim table

12. 

→ Date-dim-tbl

→ Date-id

✓ Date

✓ Full-date-description

✓ Day-of-week

Day number in calender month

Day number In calendar year

Day number in fiscal month

Day number in fiscal year

Calender-quarter

Calender-year

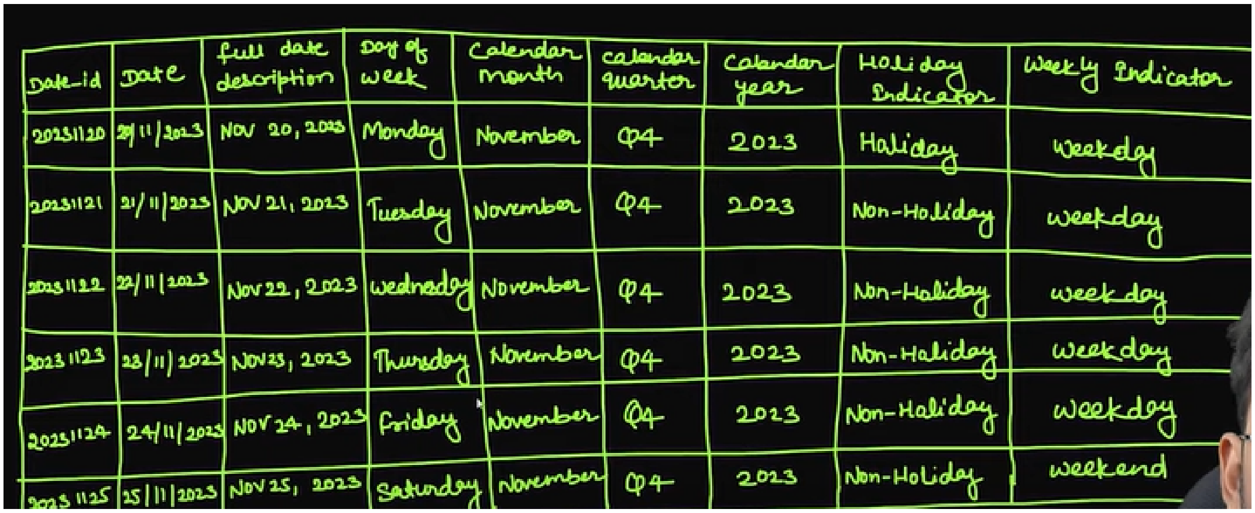
Calendar-year-month

Calendar-month-name

Haliday-indicater

Weekly-indicater

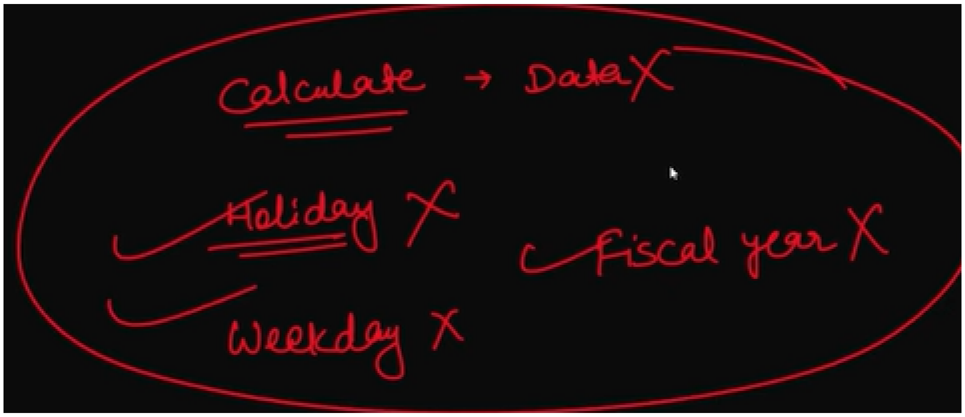
13. Lets see data in this table



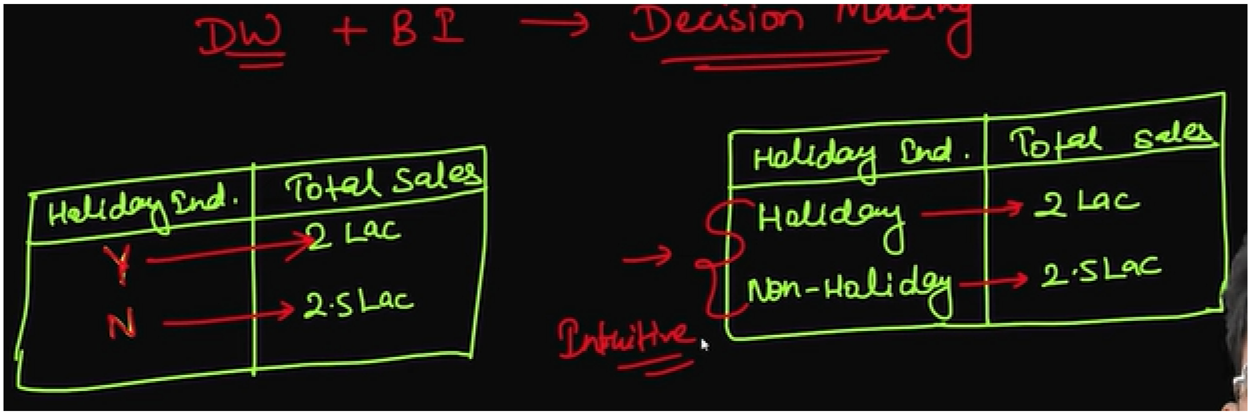
Date-id	Date	full date description	Day of week	Calendar month	calendar quarter	Calendar year	Holiday Indicator	Weekly Indicator
20231120	20/11/2023	NOV 20, 2023	Monday	November	Q4	2023	Haliday	weekday
20231121	21/11/2023	NOV 21, 2023	Tuesday	November	Q4	2023	Non-Holiday	weekday
20231122	22/11/2023	Nov 22, 2023	wednaday	November	Q4	2023	Non-Haliday	week day
20231123	23/11/2023	NOV23, 2023	Thursday	November	Q4	2023	Non-Haliday	weekday
20231124	24/11/2023	NOV 24, 2023	Friday	November	Q4	2023	Non-Haliday	weekday
20231125	25/11/2023	NOV25, 2023	Saturday	November	Q4	2023	Non-Holiday	weekend

14. Here it is on daily basis..so we'll have 7300 records

15. Here one can say..that using sql functions we can retriev day,month and year…but we cannot get this values



16. Now using joins we can easily integrate our data with date_dim table
17. So here to save the data..we can make our holiday and weekly indicatore column to boolean values
18. But in the real time…we go with what gives us more intuition


DW + BI → Decision Making

Holiday Ind.	Total Sales
Y →	2 Lac
N →	2.5 Lac

→

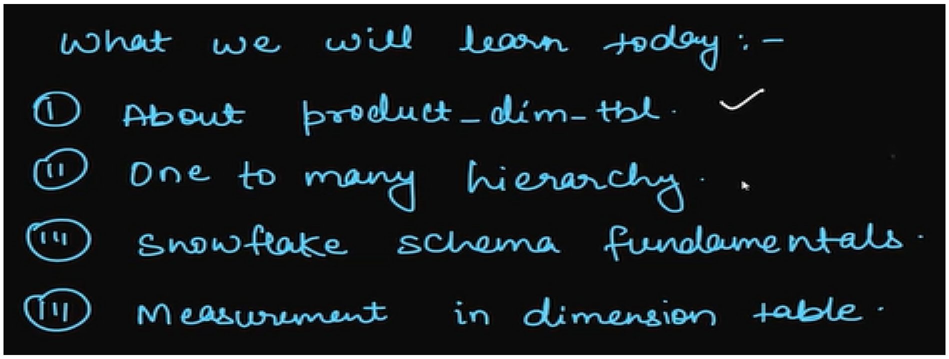
Holiday Ind.	Total sales
Holiday →	2 Lac
Non-Holiday →	2.5 Lac

Intuitive

19. So by these ..we came to know why do we need date dim table

Product Dimension and Snowflake Schema

1. So here we'll learn this today


What we will learn today :-

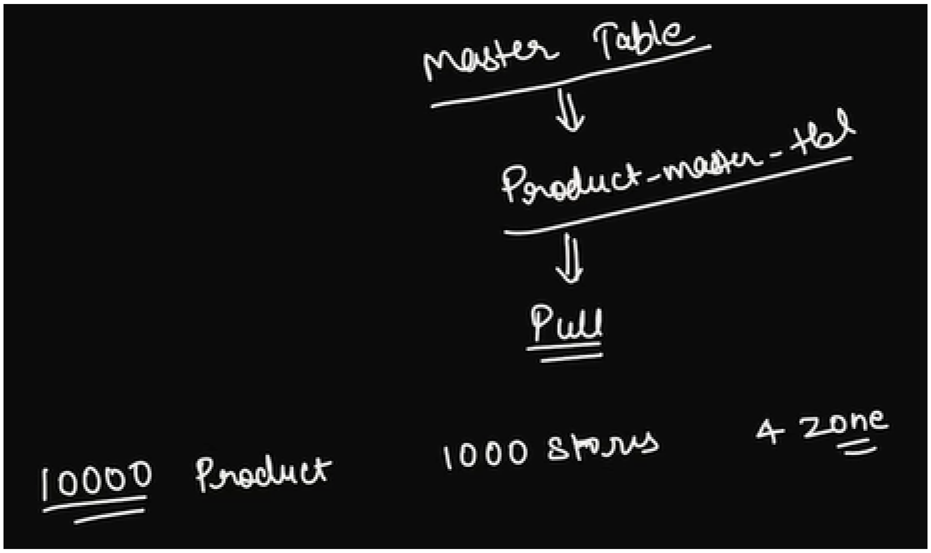
(i) About product_dim_tbl. ✓

(ii) One to many hierarchy.

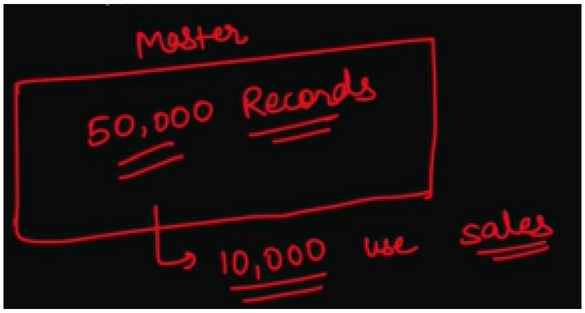
(iii) Snowflake schema fundamentals.

(iiii) Measurement in dimension table.

2. Product dim table
3. The reason y v r creating dim is that..to let BI,DA's to easily available the data and to underdtand it
4. Here lets consider we have a master table(which is a high level table and only few handle have access to it)..and next we have product_master_table

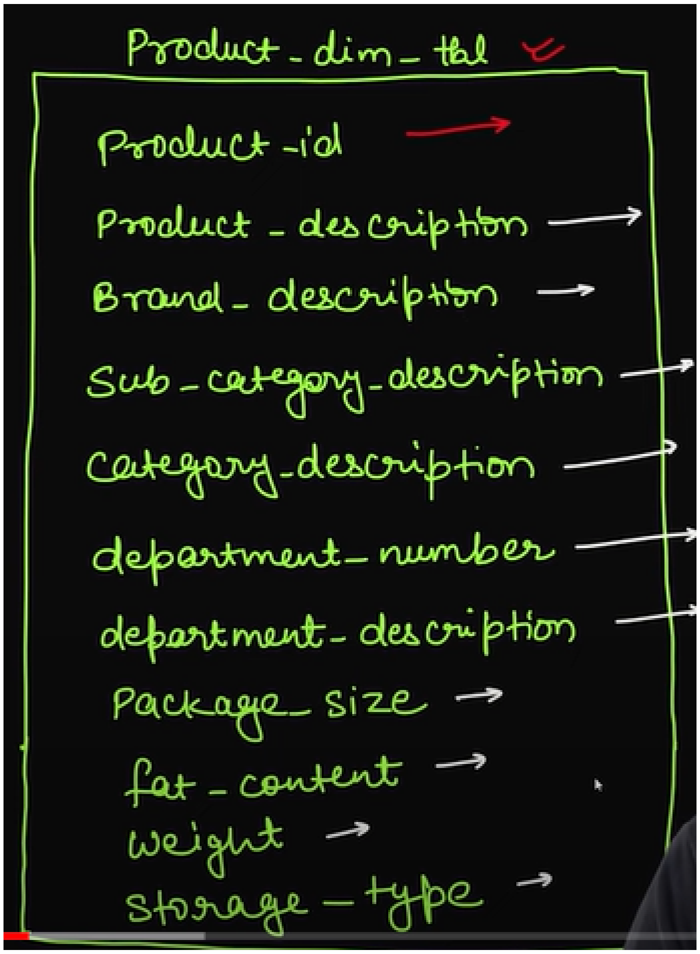
   

   from where we pull the product data
5. So in master table we cannot delete the data…it may also have extra data available

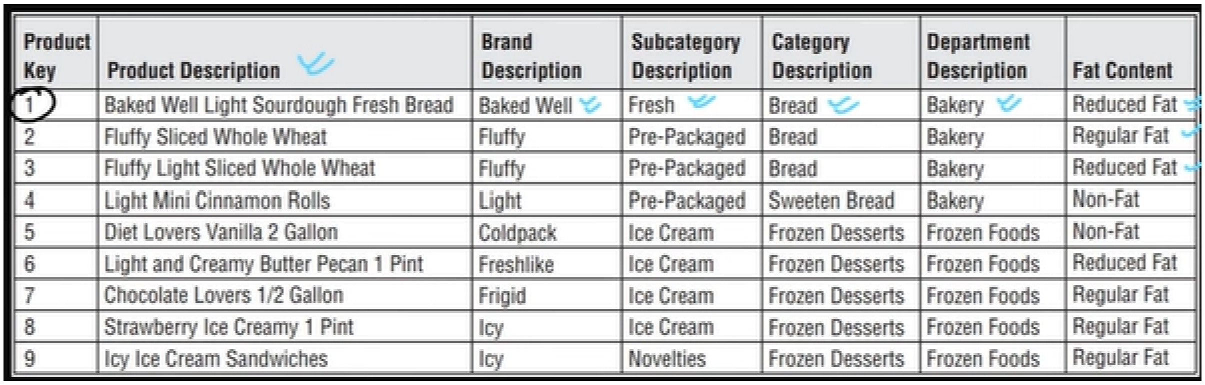
6. So here we have 4 zones right..and every zone's data will be in master table
7. And for example consider south india..here we may have data of idly and dosa…but in north zone..we dont have idly and dosa
8. Similarly here the zones tables..will get data from master for its own zone..and from zones db to stores db

9. product_dim_tbl



Product_dim_tbl

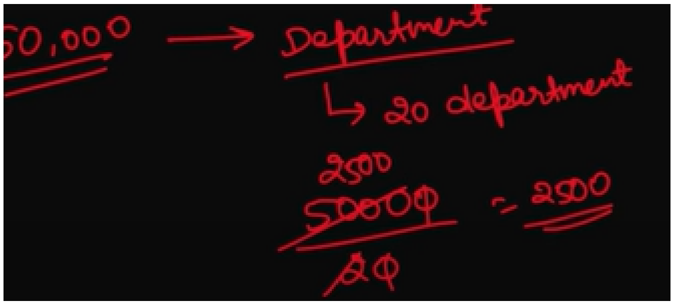
Product_id
Product_description
Brand_description
Sub_category_description
Category_description
department_number
department_description
Package_size
fat_content
weight
Storage_type



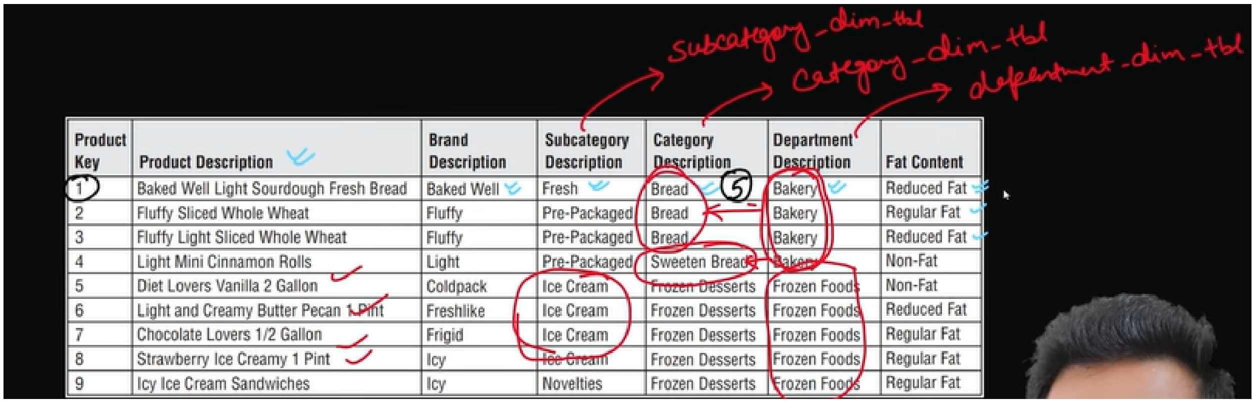
Product Key	Product Description	Brand Description	Subcategory Description	Category Description	Department Description	Fat Content
1	Baked Well Light Sourdough Fresh Bread	Baked Well	Fresh	Bread	Bakery	Reduced Fat
2	Fluffy Sliced Whole Wheat	Fluffy	Pre-Packaged	Bread	Bakery	Regular Fat
3	Fluffy Light Sliced Whole Wheat	Fluffy	Pre-Packaged	Bread	Bakery	Reduced Fat
4	Light Mini Cinnamon Rolls	Light	Pre-Packaged	Sweeten Bread	Bakery	Non-Fat
5	Diet Lovers Vanilla 2 Gallon	Coldpack	Ice Cream	Frozen Desserts	Frozen Foods	Non-Fat
6	Light and Creamy Butter Pecan 1 Pint	Freshlike	Ice Cream	Frozen Desserts	Frozen Foods	Reduced Fat
7	Chocolate Lovers 1/2 Gallon	Frigid	Ice Cream	Frozen Desserts	Frozen Foods	Regular Fat
8	Strawberry Ice Creamy 1 Pint	Icy	Ice Cream	Frozen Desserts	Frozen Foods	Regular Fat
9	Icy Ice Cream Sandwiches	Icy	Novelties	Frozen Desserts	Frozen Foods	Regular Fat

10. Here we have department description col…and each product …will falls into one of these categories

11. Lets assume we have 50000 products and 20 depts….so here each dep value will get repeated 2500 times

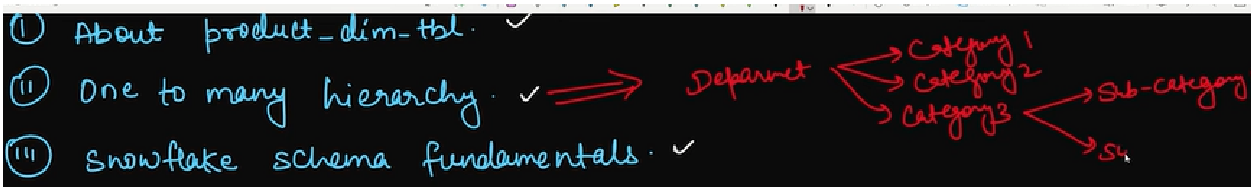


12. Here what we have seen is star schema table
13. And in snowflake schema..we'll divide this table into 3,4 tables



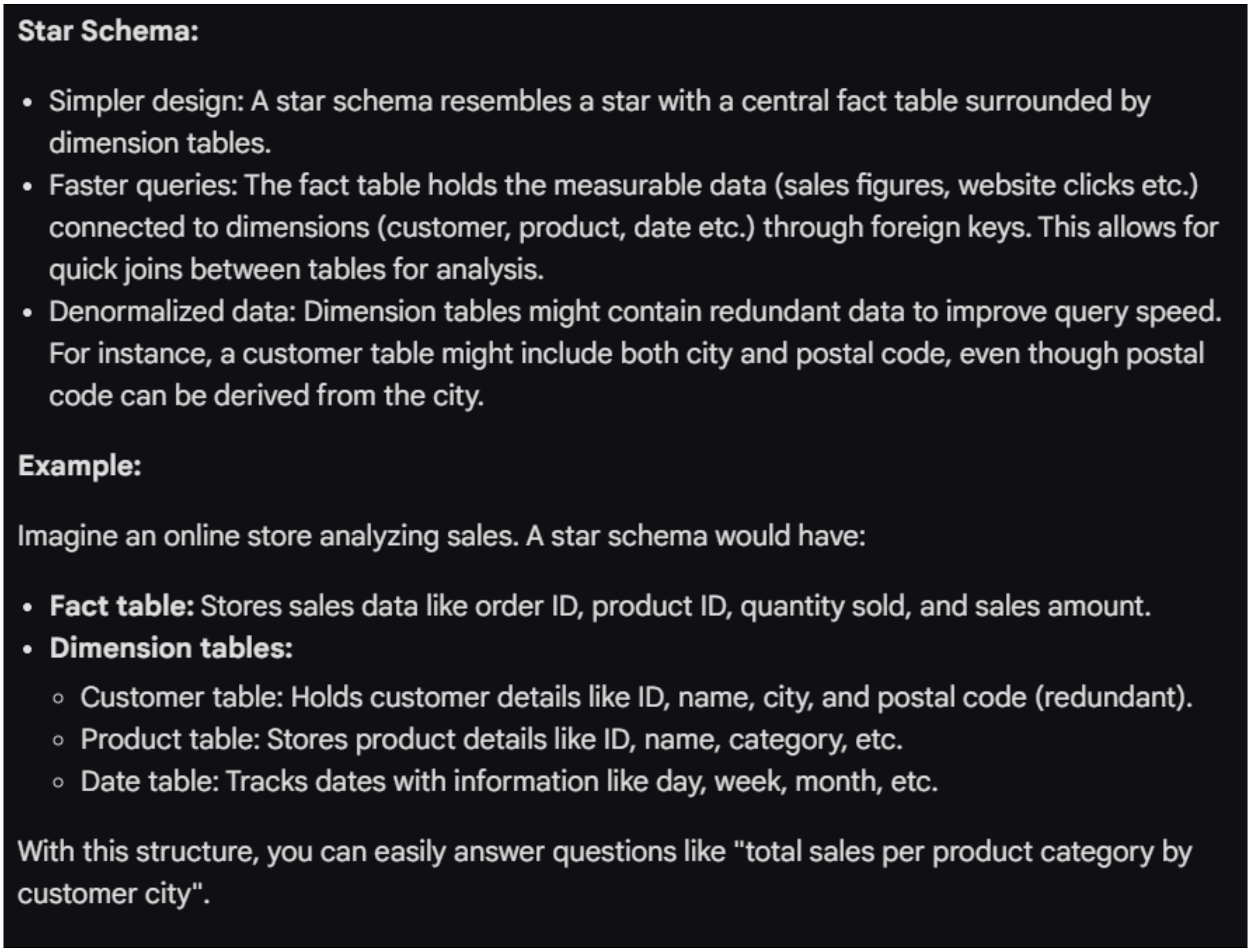
Product Key	Product Description	Brand Description	Subcategory Description	Category Description	Department Description	Fat Content
1	Baked Well Light Sourdough Fresh Bread	Baked Well	Fresh	Bread	Bakery	Reduced Fat
2	Fluffy Sliced Whole Wheat	Fluffy	Pre-Packaged	Bread	Bakery	Regular Fat
3	Fluffy Light Sliced Whole Wheat	Fluffy	Pre-Packaged	Bread	Bakery	Reduced Fat
4	Light Mini Cinnamon Rolls	Light	Pre-Packaged	Sweeten Bread	Bakery	Non-Fat
5	Diet Lovers Vanilla 2 Gallon	Coldpack	Ice Cream	Frozen Desserts	Frozen Foods	Non-Fat
6	Light and Creamy Butter Pecan 1 Pint	Freshlike	Ice Cream	Frozen Desserts	Frozen Foods	Reduced Fat
7	Chocolate Lovers 1/2 Gallon	Frigid	Ice Cream	Frozen Desserts	Frozen Foods	Regular Fat
8	Strawberry Ice Creamy 1 Pint	Icy	Ice Cream	Frozen Desserts	Frozen Foods	Regular Fat
9	Icy Ice Cream Sandwiches	Icy	Novelties	Frozen Desserts	Frozen Foods	Regular Fat

14. One to many hierarchy



(i) About product_dim_tbl. ✓
(ii) One to many hierarchy. ✓ ⟹
(iii) snowflake schema fundamentals. ✓

15. Star schema



Star Schema:

- Simpler design: A star schema resembles a star with a central fact table surrounded by dimension tables.
- Faster queries: The fact table holds the measurable data (sales figures, website clicks etc.) connected to dimensions (customer, product, date etc.) through foreign keys. This allows for quick joins between tables for analysis.
- Denormalized data: Dimension tables might contain redundant data to improve query speed. For instance, a customer table might include both city and postal code, even though postal code can be derived from the city.

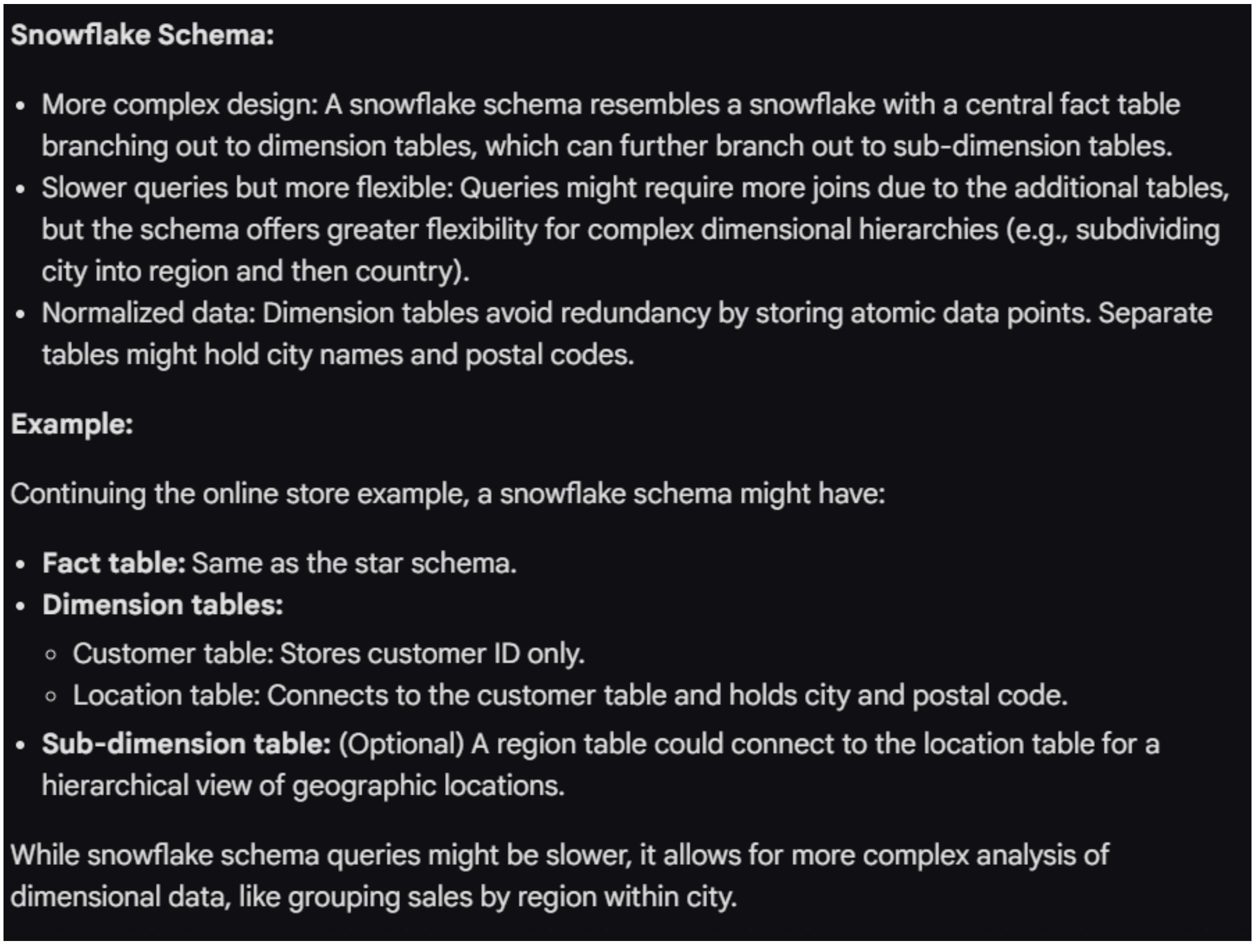
Example:

Imagine an online store analyzing sales. A star schema would have:

- **Fact table:** Stores sales data like order ID, product ID, quantity sold, and sales amount.
- **Dimension tables:**
  - Customer table: Holds customer details like ID, name, city, and postal code (redundant).
  - Product table: Stores product details like ID, name, category, etc.
  - Date table: Tracks dates with information like day, week, month, etc.

With this structure, you can easily answer questions like "total sales per product category by customer city".

16. SnowFlake Schema



Snowflake Schema:

- More complex design: A snowflake schema resembles a snowflake with a central fact table branching out to dimension tables, which can further branch out to sub-dimension tables.
- Slower queries but more flexible: Queries might require more joins due to the additional tables, but the schema offers greater flexibility for complex dimensional hierarchies (e.g., subdividing city into region and then country).
- Normalized data: Dimension tables avoid redundancy by storing atomic data points. Separate tables might hold city names and postal codes.

Example:

Continuing the online store example, a snowflake schema might have:

- **Fact table:** Same as the star schema.
- **Dimension tables:**
    - Customer table: Stores customer ID only.
    - Location table: Connects to the customer table and holds city and postal code.
- **Sub-dimension table:** (Optional) A region table could connect to the location table for a hierarchical view of geographic locations.

While snowflake schema queries might be slower, it allows for more complex analysis of dimensional data, like grouping sales by region within city.