DATA WAREHOUSE FACTS

Chapter 4

PROBLEM WITH FACTS

- Facts are usually numerical values.
- It is intuitive to add these values up.
- In Power BI analytics, we often want to add values up to totals.
- However, this is not possible for all facts, as totals do not always make sense.

FACTS-ADDITIVITY

Additive

- Fully additive facts:
  - Can be added up across all dimensions.
  - The result is meaningful and makes sense.
  - Most flexible and useful as they can be added up across all dimensions.

Semi-Additive

- Semi-additive facts:
  - Can only be added up across a few dimensions.
  - Less flexible.
  - An example will be provided later.

Non-Additive

- Non-additive facts:
  - Cannot be added up across any dimensions.
  - An example will be provided later.

ADDITIVE FACTS

product_id	name	category	sub_category
101	Product A	Category A	Subcategory A
102	Product B	Category B	Subcategory B
103	Product C	Category C	Subcategory C
104	Product D	Category D	Subcategory D
105	Product E	Category E	Subcategory E

sales_id	product_id	date_id	units	amount
1	101	20240101	5	100
2	102	20240101	5	100
3	103	20240102	8	150
4	104	20240103	12	250
5	105	20240104	20	400
6	101	20240105	10	200
7	102	20240105	15	300

65

- Example: Sales Table
  - Five different sales entries.
  - Fact: Units sold.
  - For each sale (each row), e.g., sale number three, 8 units sold.
- Adding Up Values
  - Makes sense to add up all values.
  - Total: 65 units sold.
  - This total is meaningful.

ADDITIVE FACTS MEANINGFULL?

category	Total Units Sold	
Category A	15	# 5 (20240101) + 10 (20240105)
Category B	20	# 5 (20240101) + 15 (20240105)
Category C	8	# 8 (20240102)
Category D	12	# 12 (20240103)
Category E	20	# 20 (20240104)

name	Total Units Sold	
Product A	15	# 5 (20240101) + 10 (20240105)
Product B	20	# 5 (20240101) + 15 (20240105)
Product C	8	# 8 (20240102)
Product D	12	# 12 (20240103)
Product E	20	# 20 (20240104)

date_id	Total Units Sold	
20240101	10	# 5 (product_id 101) + 5 (product_id 102)
20240102	8	# 8 (product_id 103)
20240103	12	# 12 (product_id 104)
20240104	20	# 20 (product_id 105)
20240105	25	# 10 (product_id 101) + 15 (product_id 102)

- Across All Dimensions
  - Values can be added up meaningfully across all dimensions.
- Examples of Grouping:
  - By category:
    - Grouping by category provides meaningful totals.
  - By name:
    - Grouping by name provides meaningful totals.
  - By date:
    - Adding up values across dates gives the number of units sold per day.
    - Totals remain meaningful and make sense.

SEMI-ADDITIVE FACTS

- Common Example: Account Balances
  - Day 1: Account balance of $50.
  - Day 2: Add $50, new balance is $100.
  - Day 3: No change, balance remains $100.

Day	Amount Added	Balance
1	$50	$50
2	$50	$100
3	$0	$100

SEMI-ADDITIVE FACTS

balance_id	portfolio_id	date_id	balance
1	101	20240101	10000
2	102	20240102	15000
3	107	20240103	8000
4	104	20240101	12000
5	105	20240105	20000
6	106	20240106	30000
7	107	20240107	25000

portfolio_id	portfolio_name
101	Portfolio A
102	Portfolio B

date_id	Total Balance	
20240101	22000	# Summe von 10000 (portfolio_id 101) + 12000 (portfolio_id 104)
20240102	15000	# 15000 (portfolio_id 102)
20240103	8000	# 8000 (portfolio_id 103)
20240105	20000	# 20000 (portfolio_id 105)
20240106	30000	# 30000 (portfolio_id 106)
20240107	25000	# 25000 (portfolio_id 107)

- Adding up balances across dates doesn't make sense compared to adding Up Across Portfolio Types:
- Day 1:
  - Portfolio type 1: $10000.
  - Portfolio type 2: $0.
  - Total: $10000.
- Day 2:
  - Portfolio type 1: $0.
  - Portfolio type 2: $15000.
  - Total: $15000.

General

SEMI-ADDITIVE FACTS

portfolio_name	Total Balance	
Portfolio A	22000	# Summe von 10000 (20240101) + 12000 (20240104)
Portfolio B	15000	# 15000 (20240102)
Portfolio C	8000	# 8000 (20240103)
Portfolio D	20000	# 20000 (20240105)
Portfolio E	30000	# 30000 (20240106)
Portfolio F	25000	# 25000 (20240107)

date_id	Total Balance	
20240101	22000	# Summe von 10000 (portfolio_id 101) + 12000 (portfolio_id 104)
20240102	15000	# 15000 (portfolio_id 102)
20240103	8000	# 8000 (portfolio_id 103)
20240105	20000	# 20000 (portfolio_id 105)
20240106	30000	# 30000 (portfolio_id 106)
20240107	25000	# 25000 (portfolio_id 107)

- Key Points:
  - Cannot add values across dates (doesn't make sense):
  - Adding up across dates isn't valid.
  - Dates are a typical dimension where semi-additive facts cannot be summed.

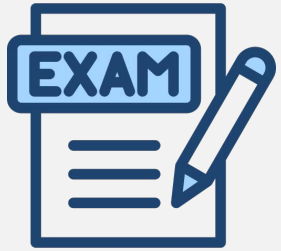
General

SEMI-ADDITIVE FACTS ALTERNATIVE APPROACH

portfolio_name	Total Balance	
Portfolio A	22000	# Summe von 10000 (20240101) + 12000 (20240104)
Portfolio B	15000	# 15000 (20240102)
Portfolio C	8000	# 8000 (20240103)
Portfolio D	20000	# 20000 (20240105)
Portfolio E	30000	# 30000 (20240106)
Portfolio F	25000	# 25000 (20240107)

date_id	Total Balance	
20240101	22000	# Summe von 10000 (portfolio_id 101) + 12000 (portfolio_id 104)
20240102	15000	# 15000 (portfolio_id 102)
20240103	8000	# 8000 (portfolio_id 103)
20240105	20000	# 20000 (portfolio_id 105)
20240106	30000	# 30000 (portfolio_id 106)
20240107	25000	# 25000 (portfolio_id 107)

- Calculate daily averages
  - Averages across dates can make sense.
  - Ensure careful usage to avoid adding semi-additive facts across all dimensions incorrectly.



NON-ADDITIVE FACTS

product_id	name	category	sub_category
101	Product A	Category A	Subcategory A
102	Product B	Category B	Subcategory B
103	Product C	Category C	Subcategory C
104	Product D	Category D	Subcategory D
105	Product E	Category E	Subcategory E

sales_id	product_id	date_id	units	price
1	101	20240101	10	200
2	102	20240102	15	300
3	103	20240103	8	150
4	104	20240104	12	250
5	105	20240105	20	400

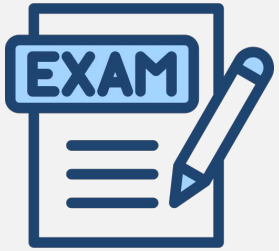
- Definition:
  - Values that cannot be meaningfully added together.
  - Example: Price of a Unit:
- Scenario:
  - Multiple units sold at different prices.
  - To find total revenue, multiply units sold by price per unit.
- Revenue:
  - Fully additive (can sum total revenue).
- Price:
  - Not additive (cannot sum prices to get a total price of a category).
  - Summing prices doesn't produce a meaningful value.

HANDLING NON-ADDITIVE FACTS

product_id	name	category	sub_category
101	Product A	Category A	Subcategory A
102	Product B	Category B	Subcategory B
103	Product C	Category C	Subcategory C
104	Product D	Category D	Subcategory D
105	Product E	Category E	Subcategory E

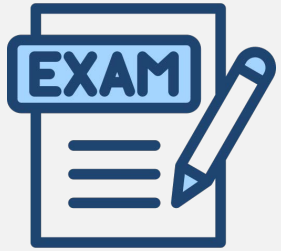
sales_id	product_id	date_id	units	price
1	101	20240101	10	200
2	102	20240102	15	300
3	103	20240103	8	150
4	104	20240104	12	250
5	105	20240105	20	400
			65	1300

- Averages:
  - Tricky because it requires considering the number of units sold.
  - Use weighted averages to get meaningful values.



TYPICAL NON-ADDITIVE FACTS

- **Prices:**
  - Cannot be summed.
- **Percentages:**
  - Usually cannot be added.
- **Ratios:**
  - Cannot be summed.
  - Example: Inventory levels in a warehouse.



NON-ADDITIVE FACTS

product_id	name	category	sub_category
101	Product A	Category A	Subcategory A
102	Product B	Category B	Subcategory B
103	Product C	Category C	Subcategory C
104	Product D	Category D	Subcategory D
105	Product E	Category E	Subcategory E

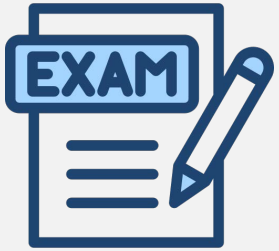
sales_id	product_id	date_id	units	price
1	101	20240101	10	200
2	102	20240102	15	300
3	103	20240103	8	150
4	104	20240104	12	250
5	105	20240105	20	400
			65	1300

- **Key Points**:
  - Non-additive facts require careful handling.
  - Adding them up in any way does not produce meaningful values.
  - Use appropriate methods like weighted averages for analysis.

DEALING WITH NON-ADDITIVE FACTS

- **Using Power BI:**
  - Create custom calculations using stored underlying values.
- **Optimizing Analytical Value:**
  - Maximize analytical insights with this approach.
- **Handling Non-Additive Facts:**
  - Lowest analytical value among fact types.
  - Requires careful consideration and appropriate handling.
- **Understanding Fact Types:**
  - Knowledge of fully additive, semi-additive, and non-additive facts is crucial.
- **Limited Analytical Value:**
  - Not suitable for straightforward aggregation.
- **Inclusion in Fact Table:**
  - Some argue against including them due to limited usefulness.
- **Better Approach:**
  - Store underlying values:
    - Example: Store both numerator and denominator for ratios.
  - Calculate resulting values in Power BI or similar tools.

General



SUMMARY

Summary of Additivity in Relation to Fact Tables

- **Fully Additive**:
  - Can sum across all dimensions.
  - Common in Transactional and Periodic Snapshot Fact Tables.
- **Semi-Additive**:
  - Can sum across some dimensions, not all (typically not across time).
  - Common in Transactional and Periodic Snapshot Fact Tables.
  - Important for measures that represent a state at a point in time (e.g., account balance).
- **Non-Additive**:
  - Cannot sum across any dimension.
  - Seen in all types of fact tables for certain measures (e.g., ratios).

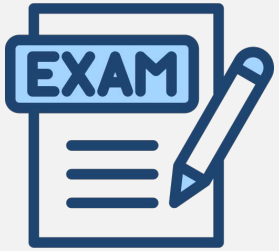
HANDLING NULL VALUES IN FACTS

Day	Incoming Amount	Average Amount
1	$50	$50
2	$100	$75
3	NULL	$75

balance_id	portfolio_id	balance	Incoming	Outgoing
1	101	10000	5000	null
2	101	15000	null	null
3	101	8000	null	null
4	102	12000	null	1000
5	102	20000	null	null

- Overview:
  - Null values can occur in facts, which is common.
- Aggregation in SQL and BI Tools:
  - Example in SQL:
    - Calculating average of incoming amounts.
    - Day 1: $50, Day 2: $100 (incoming).
    - Day 3: No incoming amount (null value).
  - Aggregation Functions:
    - SQL and BI tools (e.g., Power BI, Tableau) handle null values seamlessly.
    - Functions like average, minimum, sum can handle nulls.
  - Ease of Handling:
    - Null values are typically managed without special consideration.
    - Aggregations and calculations proceed smoothly in BI environments.

General



HANDLING NULL VALUES IN FACTS

Day	Incoming Amount	Amount for Calculation	Average Amount
1	$50	$50	$50
2	$100	$100	$75
3	NULL (treated as $0)	$0	$50

- Generally Manageable:
  - Null values are typically handled well in SQL and BI tools.
- Considerations for Averages!
- Example: Average incoming amount is $50.
  - Meaning can be misleading because not received daily.
  - Consider replacing nulls with zeros for clarity.

HANDLING NULL VALUES IN FACTS

balance_id	portfolio_id	balance	Incoming	Outgoing
1	101	10000	5000	null
2	null	15000	null	null
3	101	8000	null	null
4	null	null	null	1000
5	102	20000	null	null

balance_id	portfolio_id	balance	Incoming	Outgoing
1	101	10000	5000	0
2	999	15000	0	0
3	101	8000	0	0
4	999	0	0	1000
5	102	20000	0	0

- Null Values in Foreign Keys:
  - Issue: Null foreign keys can cause conflicts.
  - Solution: Use dummy values (e.g., 999) instead of nulls.
  - Add dummy values to dimension tables to avoid missing data issues.
- Example Scenarios:
  - Portfolio ID: If no portfolio type, use a dummy value like 999.
  - Date: Include a dummy date (e.g., 1st January 1900) if date is null.
- Connection with Dimension Tables:
  - Ensure data integrity by including dummy values.

General

YEAR-TO-DATE (YTD) FACTS: HANDLING IN DATA WAREHOUS

- Overview:
  - Specific calculations often requested by business users.
  - Includes YTD, month-to-date (MTD), quarter-to-date (QTD), etc.
- Common Approach:
  - Temptation to calculate and store these values physically in data warehouse columns.
  - Includes variations like revenue YTD or MTD.
- Problematic Nature:
  - Calculations not aligned with fact table grain (e.g., daily level).
  - Causes issues when aggregating across different dimensions and dates.
  - Leads to inaccurate aggregations and overstated values.

YEAR-TO-DATE (YTD) FACTS: HANDLING IN DATA WAREHOUS

- Recommended Approach: Store Underlying Values
  - Store base values (e.g., daily revenue) in defined grain (e.g., daily).
- Calculation in BI Tools:
  - Calculate YTD, MTD, etc., variations dynamically in BI tools (Power BI, Tableau, OLAP cubes).
- Advantages:
  - Ensures data integrity and accuracy in aggregations.
  - Avoids pitfalls of storing pre-aggregated values.

eral

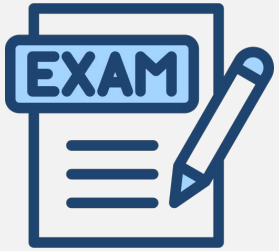
YEAR-TO-DATE (YTD) FACTS: HANDLING IN DATA WAREHOUS

- **Conclusion:**
  - **Avoid physically storing YTD calculations in data warehouse.**
  - **Calculate dynamically in BI tools to meet business needs effectively.**

eral

TRANSACTIONAL FACT TABLE
PERIODIC SNAPSHOT FACT TABLE
ACCUMULATION SNAPSHOT FACT TABLE
FACTLESS FACT TABLE

General



TRANSACTIONAL FACT TABLE

1 row = measurement of 1 event / transaction

Taken place at a specific time

One transaction defines the lowest grain

- Fundamental Fact Table Type:
  - The transactional fact table is the foundational type.
- Definition:
  - Each row corresponds to one specific event or transaction.
  - Facts are measurements associated with each event.
- Nature of Transactions:
  - Occur at a specific time and place.
  - Examples include sales transactions and other specific events.

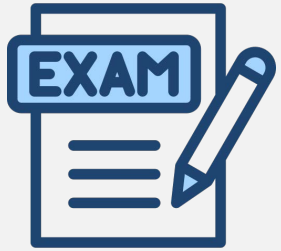
TRANSACTIONAL FACT TABLE

1 row = measurement of 1 event / transaction

Taken place at a specific time

One transaction defines the lowest grain

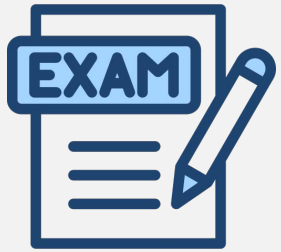
- Size and Growth:
  - Can be very large in size.
  - Rapid growth in data volume.
- Aggregation:
  - Often necessary to aggregate these tables due to their size.
- Foundation for Understanding Other Fact Tables:
  - Transactional fact table is the Fundamental type of fact table.
  - Understanding this type aids in understanding other types, such as the periodic snapshot fact table (discussed in the next lecture).



TRANSACTIONAL FACT TABLE

sales_id	product_id (FK)	date_id (FK)	units (MEASURE)
1	3	20240614	1
2	5	20240615	1
3	2	20240616	2
4	3	20240617	3
5	3	20240618	3

- **Sales Transaction:Structure**:
  - One row represents one transaction.
- **Measurements**:
  - Units sold in the transaction.
- **Foreign Keys**:
  - Product ID.
  - Time of the transaction.
  - Date and other dimensions



TRANSACTIONAL FACT TABLE

	FK	FK	FK	MEASURE
call_id	emp_id	date_id	custoer_id	duration
1	33	20240613	1	36
2	22	20240515	1	85
3	21	20240613	2	243
4	32	20240614	3	45
5	34	20240611	3	67

- **Call Event:Structure**:
  - One row represents one call event.
- **Measurements**:
  - Duration of the call.
  - Other measurements related to the call.
- **Foreign Keys**:
  - Various associated dimensions (e.g., date, time, participant IDs).

TRANSACTIONAL FACT TABLE CHARACTERISTICS

- **Common and Flexible**:
  - Most common type of fact table.
  - Highly flexible for analysis with many dimensions.
- **Additivity**:
  - Transactions and measured values are usually additive.
  - Allows for various types of aggregations and analyses.
- **Foreign Keys**:
  - Contains many foreign keys.
  - Enables detailed analysis across multiple dimensions.

TRANSACTIONAL FACT TABLE

1. Transactional Fact Table

Description:

- This type of fact table records individual transactions or events. Each row represents a single transaction at a point in time.

Additivity:

- **Fully Additive**: Measures like sales amount, quantity sold, etc., are usually fully additive across all dimensions (e.g., time, products, stores).
- **Semi-Additive**: Measures like account balances or inventory levels are often semi-additive. They can be summed across some dimensions (e.g., products, stores) but not across time.
- **Non-Additive**: Measures that do not aggregate meaningfully, like ratios or percentages.

Example:

Transaction ID	Date	Product	Quantity	Sales Amount	Inventory Balance
1	2023-06-01	A	10	$100	$500
2	2023-06-02	B	5	$50	$450

PERIODIC SNAPSHOT FACT TABLE

1 row = summarizes measure of many events / transactions

Summarized of standard period (e.g. 1 day, 1 week)

Lowest period defines the grain

- **Definition:**
  - One row is defined by the aggregation of measurements across multiple events.
- **Aggregation:**
  - Summarizes all events or transactions within a standard period.
- **Standard Period:**
  - Examples: One hour, one day, one week, one month, etc.
- **Grain Definition:**
  - The standard period (e.g., daily, weekly, hourly) defines the grain of the table.
- **Purpose:**
  - Provides a summarized view of data over regular intervals.
  - Useful for trend analysis and reporting over time.

General

EXAMPLE OF PERIODIC SNAPSHOT FACT TABLE: SALES TRANSACTIONS

MEASURE MEASURE MEASURE

week_id	revenue	sales	costs
1	572	562	10
2	345	323	22
3	689	675	14
4	321	305	16
5	124	46	78

- Underlying transactional table supports the snapshot.
- Period example: One week.
- End of the week snapshot includes aggregated measures:
  - Total revenue.
  - Number of sales.
  - Total costs.
- Characteristics:
  - Contains many measures.
  - Fewer dimensions.

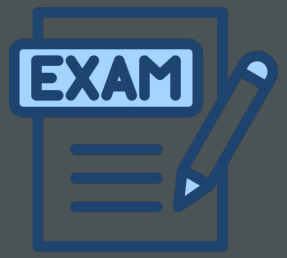
EXAMPLE OF PERIODIC SNAPSHOT FACT TABLE: CALLS EXAMPLE

MEASURE MEASURE MEASURE

day_id	no. calls	missed calls	duration
1	53	2	545
2	236	234	24
3	56	33	234
4	34	3	564
5	24	2	244

- Summarizes call data for a given period (e.g., one day).
- Measures include:
  - Number of calls.
  - Number of missed calls.
  - Total duration of calls.
- Characteristics:
  - Includes multiple measures.
  - Fewer dimensions.

General



CHARACTERISTICS OF PERIODIC SNAPSHOT FACT TABLE

- **Analytical Value**:
  - Careful grain definition retains analytical value.
  - Emphasis on measures rather than dimensions.

- **Handling Periods with No Events**:
  - Use null or zero for periods without events or transactions.
    - Use zero if it represents zero sales.
    - Use null to avoid skewing averages if no sales are expected (e.g., weekends).

CHARACTERISTICS OF PERIODIC SNAPSHOT FACT TABLE

- **Summary**
  - Periodic Snapshot Fact Tables provide controlled growth, improved performance, and maintain analytical value with a focus on measures.

PERIODIC SNAPSHOT FACT TABLE

2. Periodic Snapshot Fact Table

Description:

- This fact table captures the state of certain measures at regular intervals (e.g., daily, weekly, monthly). Each row represents a snapshot of the data at a specific point in time.

Additivity:

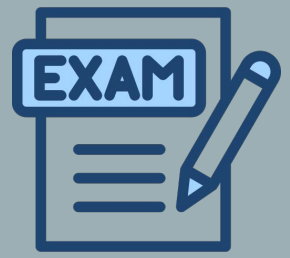
- **Fully Additive**: Aggregating across dimensions like products or stores is possible.
- **Semi-Additive**: Measures like account balances, inventory levels, and month-end positions. They can be summed across some dimensions but not across time periods.
- **Non-Additive**: Ratios or percentages.

Example:

Date	Product	Sales Amount	Inventory Level
2023-06-01	A	$100	$500
2023-06-02	A	$150	$450
2023-06-01	B	$50	$300
2023-06-02	B	$75	$275

ACCUMULATION SNAPSHOT FACT TABLE

- **General Overview**
  - Similar to the periodic snapshot fact table.
  - One row is defined by the summation of measurements of many events.
  - Period is defined by the lifespan of one process, not standardized.
  - Useful for workflow or process analysis.
- **Defining Period**:
  - Defined by the lifespan of a specific process (e.g., order fulfillment).
  - Includes a specific beginning, ending, and often milestones or steps.
- **Examples**:
  - Order fulfillment.
  - Order production process.



ACCUMULATION SNAPSHOT FACT TABLE:

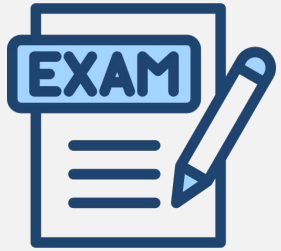
- **Benefits**:
  - Helps analyze workflows or processes.
  - Captures the entire lifecycle of a process from start to finish.
- **Usage**:
  - Ideal for tracking progress and performance of processes.
  - Enables detailed process analysis and optimization.

EXAMPLE

order_id	order_date_fk	no_products	product_fk	production_starts_fk	production_ends_fk	shipping_date_fk	inspection_date_fk	damaged_products
1001	2024-01-01	3	2001	2024-01-02	2024-01-05	2024-01-06	2024-01-07	0
1002	2024-01-03	2	2002	2024-01-04	2024-01-08	2024-01-09	2024-01-10	1
1003	2024-01-04	5	2003	2024-01-05	2024-01-09	2024-01-10	2024-01-11	2
1004	2024-01-05	1	2004	2024-01-06	2024-01-10	2024-01-11	2024-01-12	0
1005	2024-01-06	4	2005	2024-01-07	2024-01-11	2024-01-12	2024-01-13	1

- An order arrives at a manufacturer.
- Includes a date foreign key.
- Records the number and type of products ordered.
- Multiple date foreign keys for each step or milestone in the process.
  - Examples:
    - Start and end of production.
    - Date of inspection.
    - Shipping date.

- **Associated Facts**:
  - Number of damaged products found during inspectio
  - Number of products ordered.
- **Common Characteristics**:
  - Contains measures related to each step.
  - Many date dimensions.
  - Typical to have numerous date dimensions in the accumulation snapshot fact table.



ACCUMULATION SNAPSHOT FACT TABLE: CHARACKTERSITICA

Date FK	Measure	Prod FK	Date FK	Date FK	Date FK	Date FK	Measure
order_id	order_date_fk	no_products	product_fk	production_starts_fk	production_ends_fk	shipping_date_fk	inspection_date_fk	damaged_products
1001	2024-01-01	3	2001	2024-01-02	2024-01-05	2024-01-06	2024-01-07	0
1002	2024-01-03	2	2002	2024-01-04	2024-01-08	2024-01-09	2024-01-10	1
1003	2024-01-04	5	2003	2024-01-05	2024-01-09	2024-01-10	2024-01-11	2
1004	2024-01-05	1	2004	2024-01-06	2024-01-10	2024-01-11	2024-01-12	0
1005	2024-01-06	4	2005	2024-01-07	2024-01-11	2024-01-12	2024-01-13	1

- **Usage**: Least common type but important for workflow and process analysis.
- **Process Analysis**: Particularly useful when analyzing workflows or processes.
- **Date Foreign Keys**: Many date or time foreign keys for each process step.
- **Role-Playing Dimension**: Typically uses a single dimension table for multiple connections, known as a role-playing dimension.

PERIODIC SNAPSHOT FACT TABLE

3. Accumulation Snapshot Fact Table

Description:

- This type of fact table captures the cumulative state of a process at various stages. It shows the progress of a workflow or a pipeline over time.

Additivity:

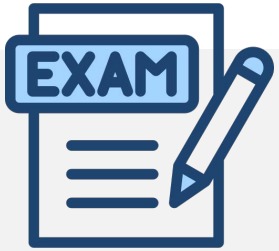
- **Fully Additive**: Often not applicable since the table is designed to track cumulative progress rather than summable transactions.
- **Semi-Additive**: Useful for measures like total amount billed, total amount received, etc., which can be summed across stages but not over time.
- **Non-Additive**: Similar measures as above, where meaningful aggregation over certain dimensions might not apply.

Example:

Order ID	Order Date	Shipped Date	Billed Amount	Received Amount
1	2023-06-01	2023-06-05	$200	$200
2	2023-06-02	2023-06-06	$150	$100

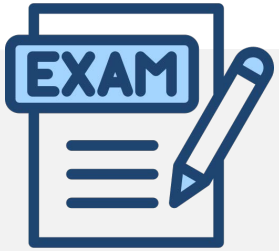
ACCUMULATION SNAPSHOT FACT TABLE SUMMARY

Summary: Understanding the usage of a single dimension across multiple connections.



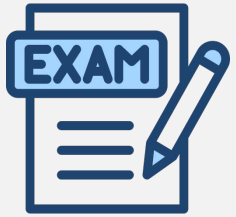
COMPARISON OF FACT TABLES:

- **Transactional Fact Table**:
  - **Definition**: One row per transaction.
  - **Dimensions**: Highly dimensional, aiding detailed analysis.
  - **Date Dimension**: Transaction date.
  - **Measurement of Facts**: Performance of individual transactions.
  - **Characteristics**: Largest size due to detailed grain, flexible for analysis.
  - **Performance**: Sometimes requires aggregation for optimal performance.



COMPARISON OF FACT TABLES:

- **Periodic Snapshot Fact Table**:
  - **Definition**: One row per aggregated period.
  - **Dimensions**: Fewer dimensions compared to transactional, more aggregated data.
  - **Date Dimension**: Snapshot date, end of the period.
  - **Measurement of Facts**: Cumulative measurement over defined periods.
  - **Characteristics**: Moderately sized due to aggregated nature.
  - **Performance**: Generally better due to pre-aggregated data.



COMPARISON OF FACT TABLES:

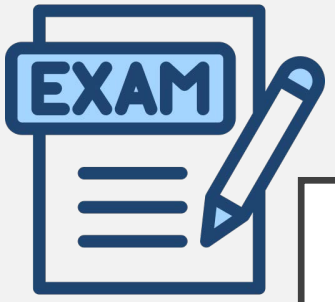
- **Accumulating Snapshot Fact Table**:
  - **Definition**: One row per process or event lifespan.
  - **Dimensions**: Many date foreign keys, high dimensional.
  - **Date Dimension**: Multiple snapshot dates related to process milestones.
  - **Measurement of Facts**: Measurement across the lifetime of processes.
  - **Characteristics**: High dimensionality due to many date foreign keys.
  - **Performance**: Can be complex due to high dimensionality.

COMPARISON OF FACT TABLES: OVERALL ANALYSIS:

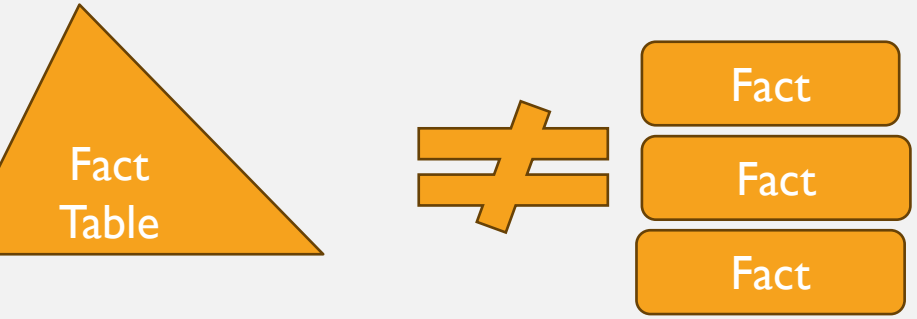
- **Transactional vs. Snapshot**:
  - Transactional: Detailed, flexible but potentially larger.
  - Snapshot: Aggregated, better performance with defined periods.
  - **Performance Considerations**:
  - Aggregation helps in improving performance across all types.
  - Modern databases mitigate many performance concerns.

COMPARISON OF FACT TABLES: OVERALL ANALYSIS
CONCLUSION

- Understanding the nuances and applications of each fact table type is crucial for effective data modeling and analysis in business intelligence.



FACTLESS FACT TABLE



- Definition:
  - A type of fact table that records only dimensional information without any measurable facts.
- Concept:
  - While a fact table typically contains numerical measurements (facts), a factless fact table focuses solely on capturing relationships between dimensions.

FACTLESS FACT TABLE

id	entrydate	Lob_id	Manger_id	Pos_id
	20210503	2	4	23
	20210321	2	3	21
	20240512	3	4	23
	20231201	1	1	19
	20211130	1	1	16

Emplyee registration

- Characteristics:
  - Purpose: Tracks dimensional aspects of events or transactions without numerical measurements.
  - Contents: Contains only foreign keys and dimensional attributes.
- Examples: Used when analyzing events or transactions based on relationships rather than numerical data.
- Usage: Helpful in scenarios where understanding relationships or occurrences between dimensions is critical, such as in event tracking or process analysis.

FACTLESS FACT TABLE
EXAMPLE SCENARIO: RECORDING NEW EMPLOYEE REGISTRATIONS IN A COMPANY.

id	entrydate	Lob_id	Manger_id	Pos_id
	20210503	2	4	23
	20210321	2	3	21
	20240512	3	4	23
	20231201	1	1	19
	20211130	1	1	16

Emplyee registration

- Table Contents:
  - Dimensional Aspects: Entry date, department, region, manager, and other associated attributes.
- Metrics:
  - None; the table focuses solely on dimensional details.
- Purpose:
  - Captures every new employee registration event.
  - Enables querying and analysis based on dimensional attributes rather than numerical metrics.

FACTLESS FACT TABLE
EXAMPLE SCENARIO: RECORDING NEW EMPLOYEE REGISTRATIONS IN A COMPANY.

id	entrydate	Lob_id	Manger_id	Pos_id
	20210503	2	4	23
	20210321	2	3	21
	20240512	3	4	23
	20231201	1	1	19
	20211130	1	1	16

Emplyee registration

- Use Cases
  - Tracking Events: Answers questions such as how many employees were registered last month.
- Dimensional Analysis
  - Allows filtering and counting based on dimensions like region or department.
- Implementation
  - Utilizes SQL queries or BI tools to filter by dimensions (e.g., date, region) and count rows to derive insights.

FACTLESS FACT TABLE
EXAMPLE SCENARIO: RECORDING NEW EMPLOYEE REGISTRATIONS IN A COMPANY.

id	entrydate	Lob_id	Manger_id	Pos_id
	20210503	2	4	23
	20210321	2	3	21
	20240512	3	4	23
	20231201	1	1	19
	20211130	1	1	16

Emplyee registration

- Benefits:
  - Provides a structured way to analyze events and their dimensional aspects.
  - Demonstrates that a fact table can exist without traditional numerical metrics, focusing instead on relationships and occurrences.

FACTLESS FACT TABLE
EXAMPLE SCENARIO: RECORDING PROMOTIONS WITHOUT ASSOCIATED METRICS.

id	entrydate	prod_id	channel_id	campaign_id
	20210503	2	4	23
	20210321	2	3	21
	20240512	3	4	23
	20231201	1	1	19
	20211130	1	1	16

- Table Contents:
  - Dimensional Aspects: Promo code, product promoted, campaign details, and other associated attributes.
  - Metrics: None; focuses on capturing event occurrences and dimensional details.
  - Purpose:
    - Tracks promotional events and their associated dimensional aspects.
    - Enables analysis based on event occurrences rather than numerical metrics.

FACTLESS FACT TABLE
EXAMPLE SCENARIO: RECORDING PROMOTIONS WITHOUT ASSOCIATED METRICS.

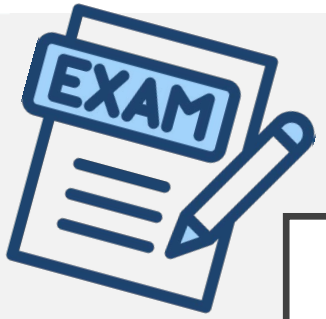
id	entrydate	prod_id	channel_id	campaign_id
	20210503	2	4	23
	20210321	2	3	21
	20240512	3	4	23
	20231201	1	1	19
	20211130	1	1	16

- Use Cases:
- **Event Tracking:** Records each promotion event without measuring specific metrics like sales or revenue.
- **Dimensional Analysis**: Facilitates analysis based on promo codes, products, or campaigns.
- **Implementation**:
  - Structured in a fact table format to organize and query event occurrences.
  - Queries can filter and analyze based on dimensional attributes using SQL or BI tools.

FACTLESS FACT TABLE
EXAMPLE SCENARIO: RECORDING PROMOTIONS WITHOUT ASSOCIATED METRICS.

id	entrydate	prod_id	channel_id	campaign_id
	20210503	2	4	23
	20210321	2	3	21
	20240512	3	4	23
	20231201	1	1	19
	20211130	1	1	16

- **Benefits**:
  - Provides a comprehensive view of promotional activities and their relationships.
  - Demonstrates the flexibility of a fact table to capture events even without numerical metrics.



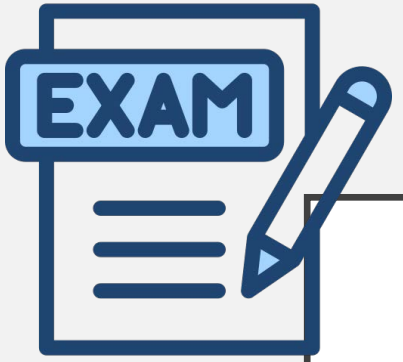
DESIGNING A FACT TABLE: KEY DECISIONS

What are the key actions to perform tp create fact tabels?

4 decissions

Create tables

- **Identify Grain**:
  - Determine the level of detail for each row in the fact table.
  - Answer questions like: Should each row represent a daily transaction, weekly aggregation, or another defined period?
- **Select Measures**:
  - Define which numerical metrics or measures to include in the fact table.
  - Consider metrics relevant to business analysis and reporting needs.
- **Choose Dimensions**:
  - Decide on the dimensions that provide context to the measures.
  - Dimensions could include time, location, product, customer, etc.
- **Handling Nulls and Missing Data**:
  - Decide how to handle null values or missing data in the fact table.
  - Consider whether to use actual zeros, nulls, or placeholder values.



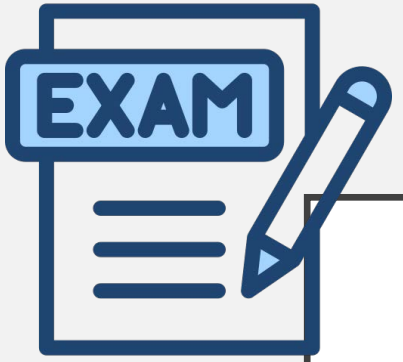
DESIGNING A FACT TABLE: KEY DECISIONS

What are the key actions to perform tp create fact tabels?

4 decissions

Create tables

- **Decision-Making Process**:
  - **Business Needs**:
    - Base decisions on specific business requirements and analytical goals.
    - Ensure the design aligns with how the data will be analyzed and used in reporting.
  - **Implementation**:
    - Translate decisions into the actual table structure and column definitions.
    - Ensure the design supports efficient querying and performance in BI tools or SQL.



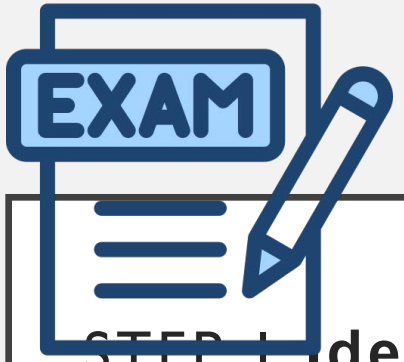
DESIGNING A FACT TABLE: KEY DECISIONS

What are the key actions to perform tp create fact tabels?

4 decissions

Create tables

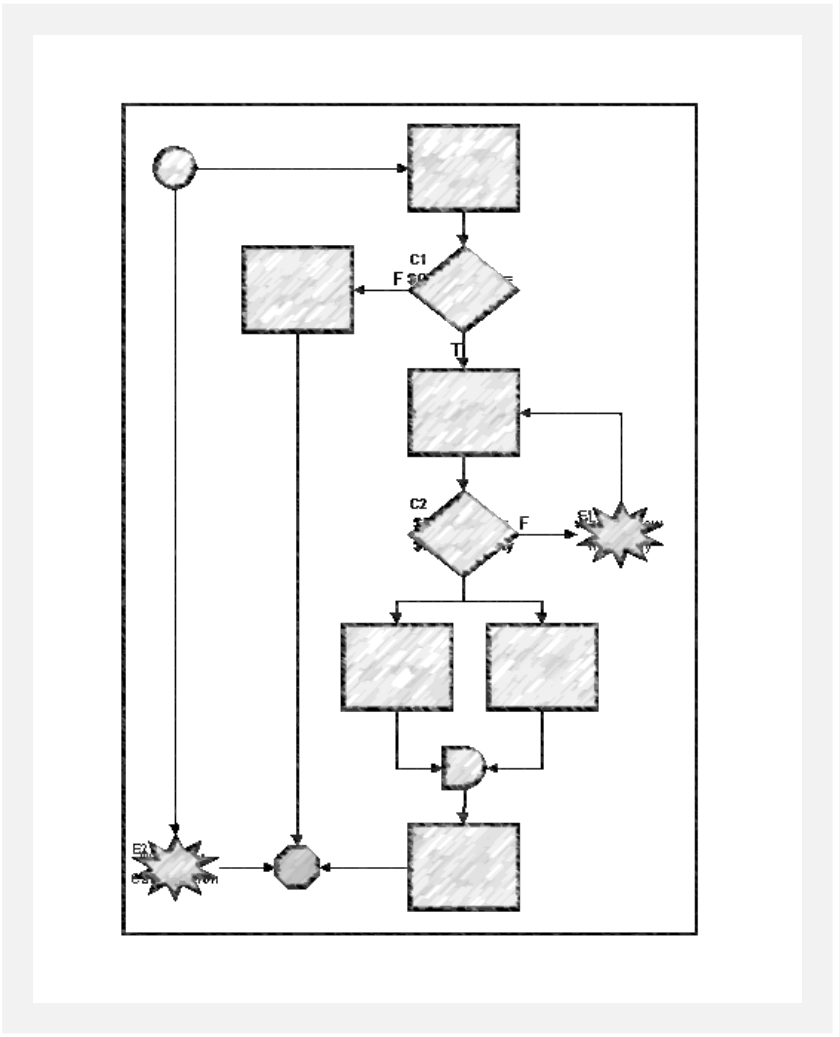
- **Practical Application**:
  - By answering these key questions, we can effectively design a fact table that meets business objectives and facilitates meaningful analysis and reporting.



DETAILS

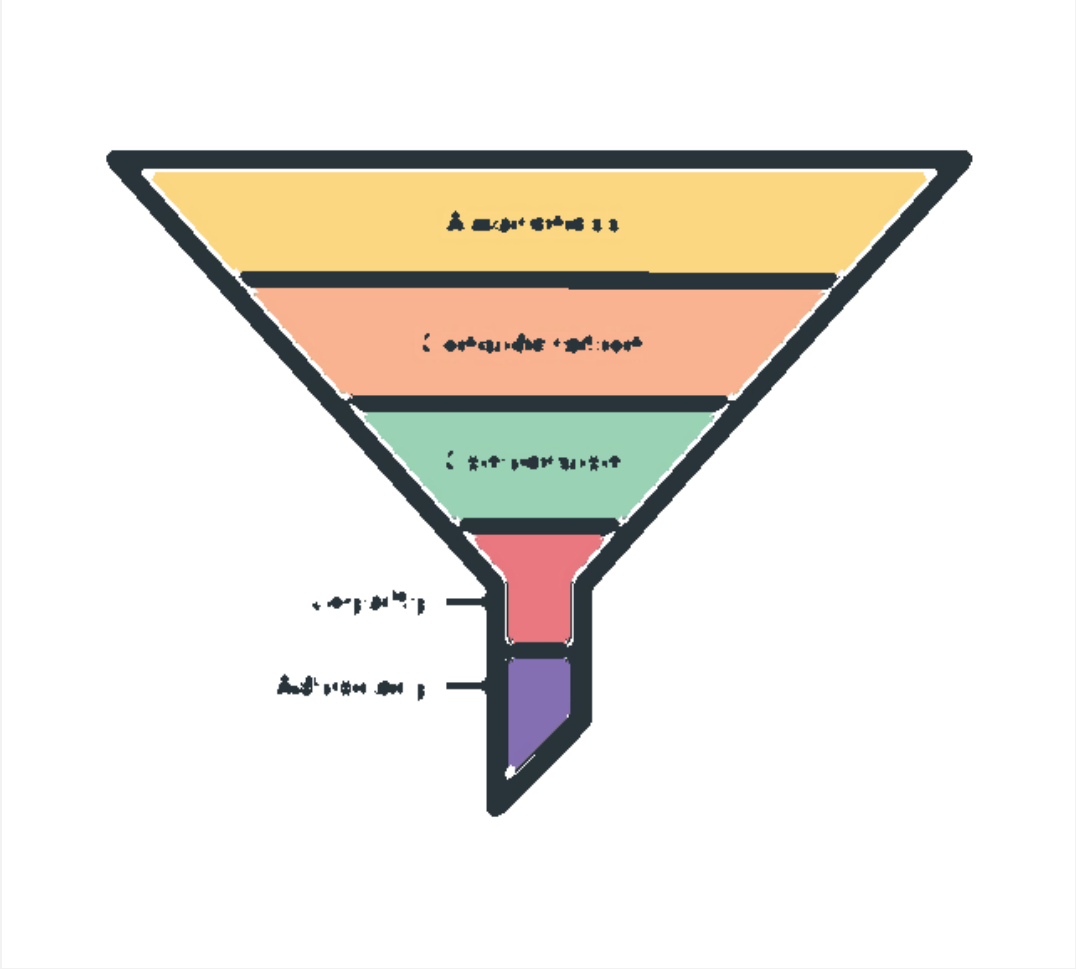
STEP 1 Identify Business Process

- Determine the specific business process that requires analysis, such as sales, order processing, or fulfillment.
- Example: Creating a sales fact table to analyze sales transactions.



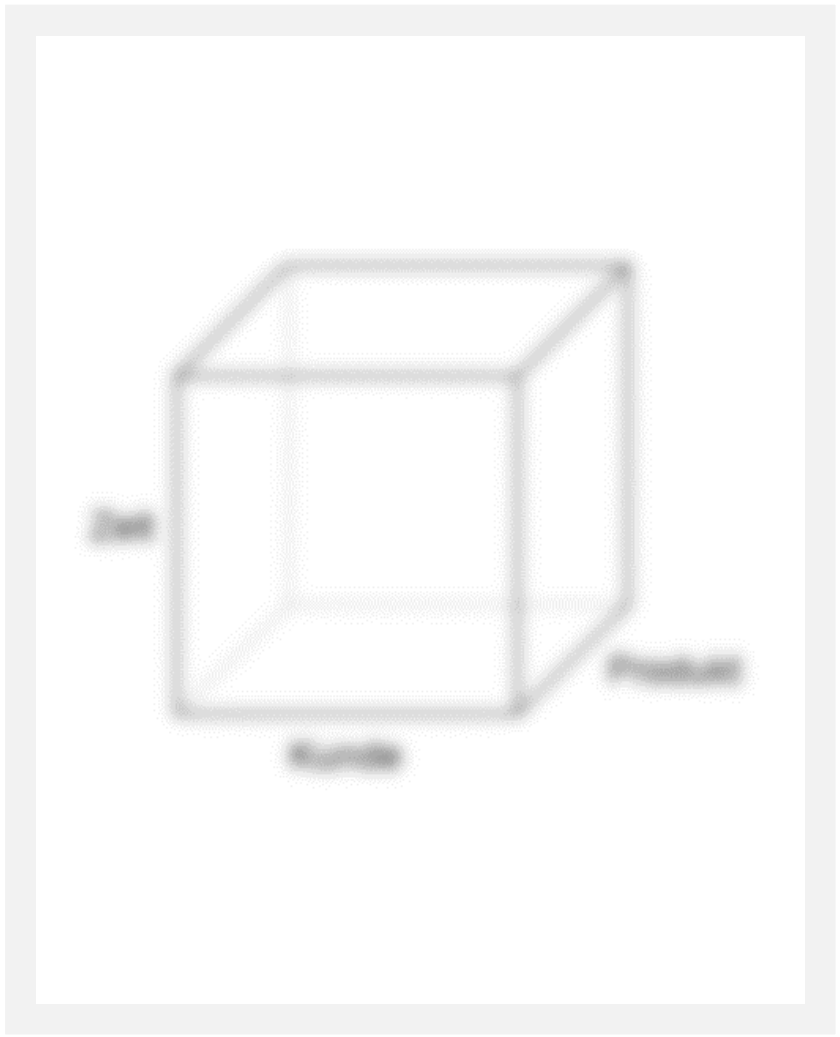
DETAILS
STEP II

- **Define Grain**:
  - Decide on the level of detail or granularity for each row in the fact table.
  - The grain defines what each row represents, such as a single transaction or a daily aggregation.
  - Opt for a finer grain to retain flexibility and avoid limiting future analysis options with pre-aggregated data.



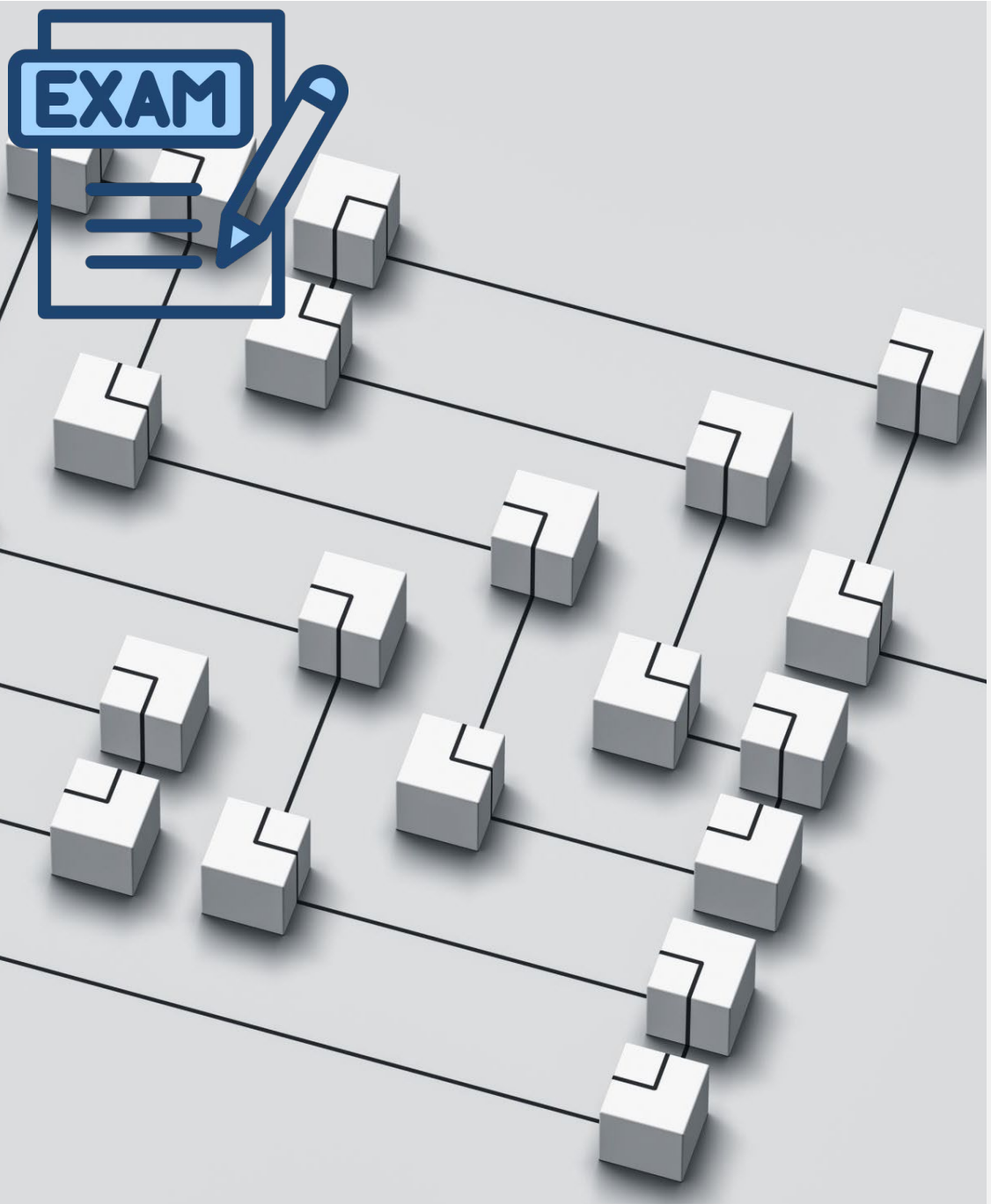
DETAILS
STEP III

- **Identify Relevant Dimensions**:
  - Identify dimensions that provide context to the measures in the fact table.
  - Dimensions include aspects like time (when), location (where), product (what), customer (who), etc.
  - Dimensions are crucial for filtering, grouping, and analyzing data, often referred to as the "soul" of data analysis.



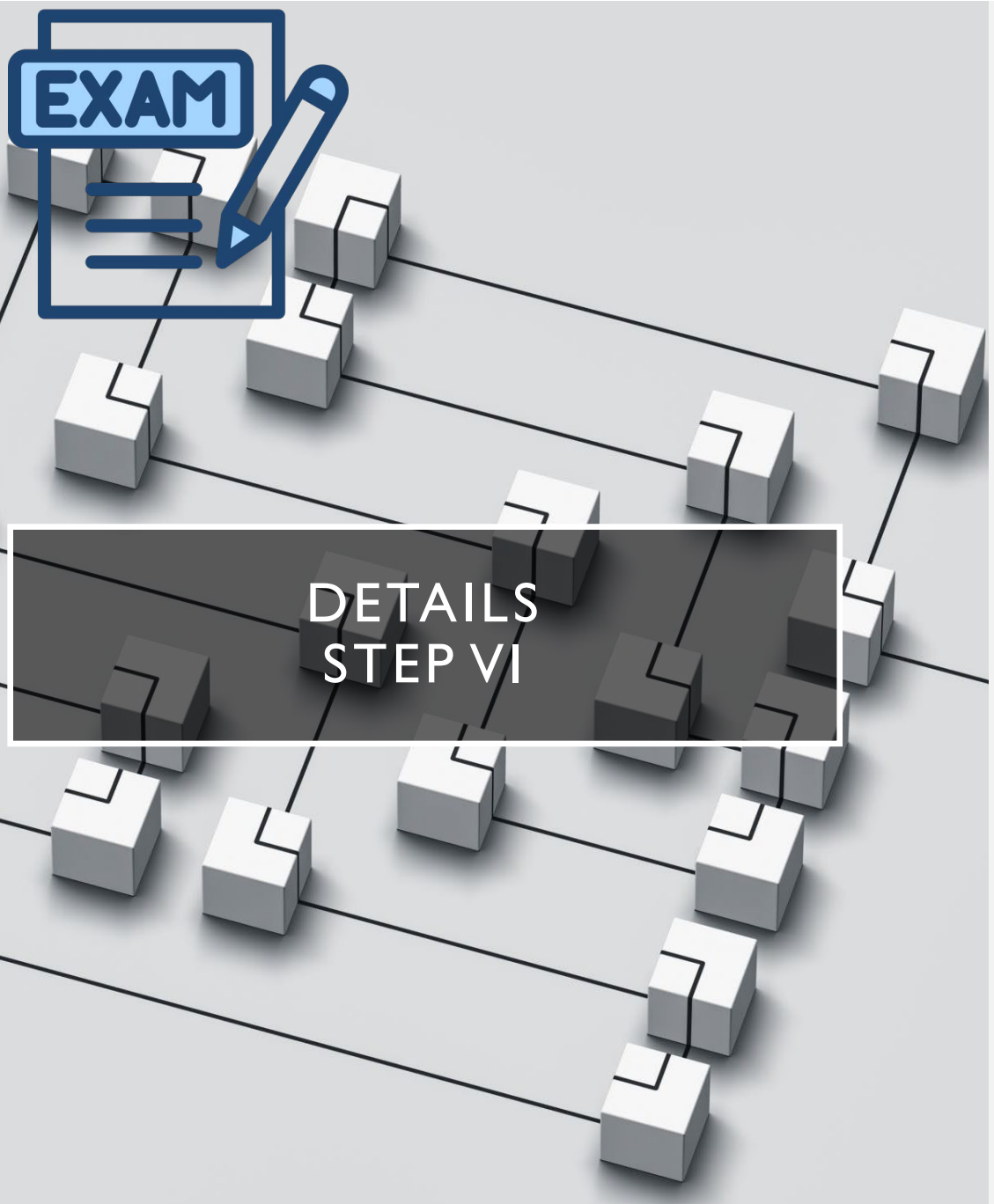
DETAILS
STEP IV

- **Identify Facts (Measurements)**:
  - Determine the numerical measurements or metrics (facts) that will be stored in the fact table.
  - These are the quantifiable data points that will be analyzed and aggregated as per the defined grain.
  - Example: Revenue, quantity sold, cost, etc.



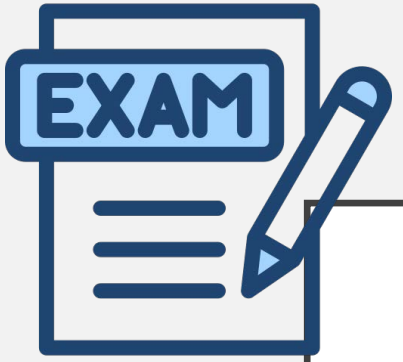
DETAILS
STEP V

- **Aggregation Strategy**:
  - Decide if aggregation of facts is necessary based on the defined grain.
  - Aggregations may be needed for reporting or specific analytical purposes in data marts.
  - Maintain flexibility by starting with atomic transactional data before any aggregation.



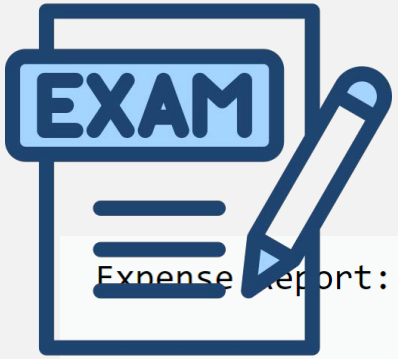
DETAILS
STEP VI

- **Consider ETL Aspects**:
  - While designing, keep in mind the Extraction, Transformation, and Loading (ETL) processes.
  - Ensure the design supports efficient data extraction, transformation for analysis, and loading into the fact table.



DETAILS
CONCLUSION

- Following these steps ensures a well-designed fact table that aligns with business needs and facilitates comprehensive data analysis.
- These steps provide a structured approach to designing fact tables, laying the foundation for effective business intelligence and reporting capabilities.



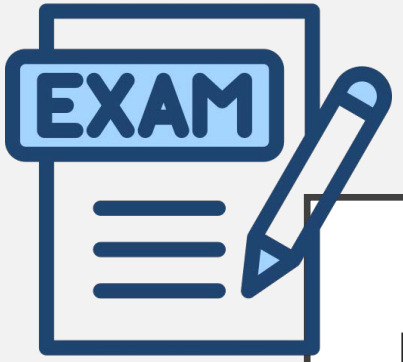

```
Expense Report:  ER-123

Timesheets:      TS-123

AR Invoices:     INV-123
```

d	order_date	no_products	product_name	production_starts	production_ends	shipping_date
	2024-01-01	3	Widget A	2024-01-02	2024-01-05	2024-01-06
	2024-01-03	2	Gadget B	2024-01-04	2024-01-08	2024-01-09
	2024-01-04	5	Thingamajig C	2024-01-05	2024-01-09	2024-01-10
	2024-01-05	1	Doohickey D	2024-01-06	2024-01-10	2024-01-11
	2024-01-06	4	Gizmo E	2024-01-07	2024-01-11	2024-01-12

D	Customer	SalesValue	Product	Date
1	Han	100	Engine	28/04/2015
2	Anakin	200	Computer	28/04/2015
3	Luke	300	Pencil	28/04/2015

NATURAL KEYS DEFINITION

- **Definition:** Natural keys are inherent identifiers that originate from the source system.
- **Example:** In a product dimension, the product ID might be an alphanumeric number generated by the system.
- **Characteristics**
  - Can be bulky and not user-friendly.
  - Often large in size, potentially causing inefficiencies in handling and storage.
  - Directly reflect the key as it is in the source system, like transaction IDs in transactional tables.



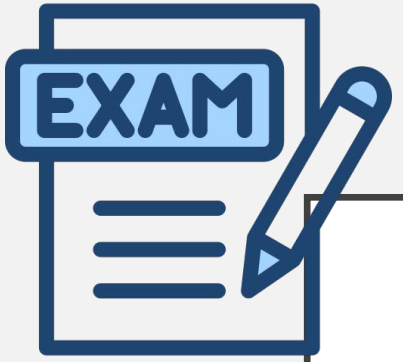
NATURAL KEYS
DISADVANTAGES OF NATURAL KEYS:

```
Expense Report: ER-123

Timesheets:     TS-123

AR Invoices:    INV-123
```

d	order_date	no_products	product_name	production_starts	production_ends	shipping_date
	2024-01-01	3	Widget A	2024-01-02	2024-01-05	2024-01-06
	2024-01-03	2	Gadget B	2024-01-04	2024-01-08	2024-01-09
	2024-01-04	5	Thingamajig C	2024-01-05	2024-01-09	2024-01-10
	2024-01-05	1	Doohickey D	2024-01-06	2024-01-10	2024-01-11
	2024-01-06	4	Gizmo E	2024-01-07	2024-01-11	2024-01-12

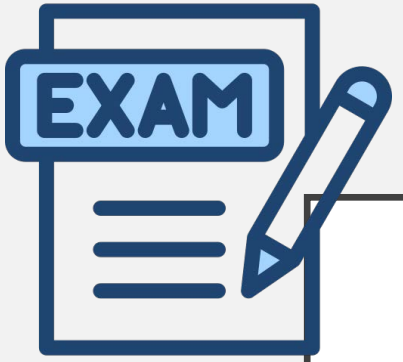
ID	Customer	SalesValue	Product	Date
1	Han	100	Engine	28/04/2015
2	Anakin	200	Computer	28/04/2015
3	Luke	300	Pencil	28/04/2015

- **Complexity:** They may not be intuitive or easy to manage due to their format and size.
- **Storage Inefficiency:** Larger size can lead to increased storage requirements.
- **Limited Flexibility:** They tie the data warehouse structure closely to the source system's format, limiting flexibility in data management and integration.



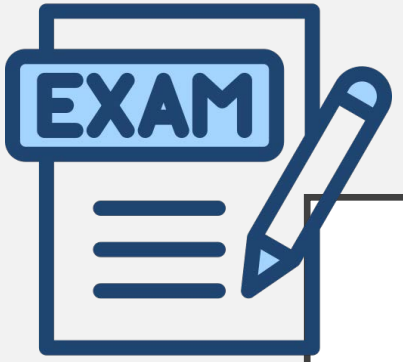
SURROGATE KEYS

- **Definition**: Surrogate keys are system-generated unique identifiers specifically designed for use within the data warehouse.
- **Purpose**: Designed to address the limitations of natural keys by providing simpler, smaller, and more manageable identifiers.
- **Example**: A sequentially assigned integer key for products or transactions within the data warehouse.



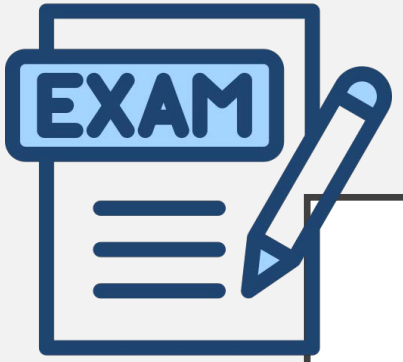
ADVANTAGES OF SURROGATE KEYS

- **Simplicity**: They are typically simple integers or numeric identifiers.
- **Compactness**: Surrogate keys are smaller in size compared to natural keys, enhancing efficiency in storage and processing.
- **Flexibility**: Decouples the data warehouse structure from the source system, allowing for easier data integration and management.



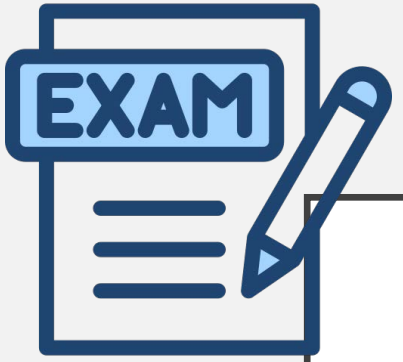
SURROGATE KEYS: GUIDELINES FOR USAGE:

- **Prefer Surrogate Keys**: Use surrogate keys over natural keys to improve performance, simplify data management, and enhance flexibility.
- **Consistency**: Ensure surrogate keys are consistently applied across related tables for proper data relationships.
- **Mapping**: Maintain mapping between natural keys from source systems and surrogate keys in the data warehouse for reference and integration purposes.



BEST PRACTISE

1. **Use Integer Data Types:**
   - Surrogate keys should be simple integers, typically `INT` or `BIGINT`. This ensures they are small, fast to compare, and efficient to store.
2. **Auto-Incremented Values:**
   - Surrogate keys should be auto-incremented values generated by the database management system (DBMS). This ensures uniqueness and simplicity in key generation.
3. **Avoid Meaningful Data:**
   - Surrogate keys should have no business meaning. They are purely for identifying rows uniquely within a table. Avoid embedding business logic or semantics in the surrogate key.



BEST PRACTISE

4. **Consistent Use Across Tables:**
   - Use surrogate keys consistently across all related tables. For instance, if you have a `product` dimension table with a surrogate key, use this key in all fact tables referencing `product`.
5. **Primary Keys and Indexes:**
   - Always define surrogate keys as primary keys. This not only enforces uniqueness but also optimizes query performance through indexing.
6. **Foreign Key Relationships:**
   - Use surrogate keys as foreign keys in related tables to maintain referential integrity. This simplifies joins and ensures data consistency.

BEST PRACTISE

7. **Single Column Keys:**
    - Surrogate keys should be single-column keys. Composite keys can complicate query design and indexing.
8. **Non-Null and Unique Constraints:**
    - Ensure surrogate keys are non-null and unique. This is typically handled automatically when they are defined as primary keys with auto-increment properties.
9. **Surrogate Key Generation:**
    - Let the DBMS handle the generation of surrogate keys. Most modern databases like MySQL, PostgreSQL, SQL Server, and Oracle support auto-increment or sequence-based key generation.
10. **Keep Natural Keys:**
    - While surrogate keys are used for primary and foreign key relationships, it's often useful to

NATURAL KEYS VS. SURROGATE KEYS

order_id	order_date	no_products	product_name	production_starts	production_ends	shipping_date
1001	2024-01-01	3	Widget A	2024-01-02	2024-01-05	2024-01-06
1002	2024-01-03	2	Gadget B	2024-01-04	2024-01-08	2024-01-09
1003	2024-01-04	5	Thingamajig C	2024-01-05	2024-01-09	2024-01-10
1004	2024-01-05	1	Doohickey D	2024-01-06	2024-01-10	2024-01-11
1005	2024-01-06	4	Gizmo E	2024-01-07	2024-01-11	2024-01-12

order_id	order_date_sk	no_products	product_sk	production_starts_sk	production_ends_sk
1	101	3	201	301	401
2	102	2	202	302	402
3	103	5	203	303	403
4	104	1	204	304	404
5	105	4	205	305	405

PRACTICAL CASE STUDY FOR FACT TABLES E-COMMERCE

Brainstorming

PRACTICAL CASE STUDY FOR FACT TABLES
E-COMMERCE
CASE STUDY OVERVIEW

- **Scenario**: Corporate IT of a large e-commerce company.
- **Business Operations**: Selling multiple products across three large websites, each managed independently with multiple departments.
- **Product Range**: Approximately 1,000 individual products including groceries, kitchen products, and household items.

Brainstorming

PRACTICAL CASE STUDY FOR FACT TABLES
E-COMMERCE
DATA COLLECTION POINTS

- **Data Collection Points:**

1. **Shopping Cart Checkout**:
    1. **Data Collected**:
        1. Customer name
        2. Order ID
        3. Order line ID
        4. Quantity
        5. Unit price
        6. Sales amount
    2. **Granularity**: Each row represents a single item order line within an order.
    3. **Example**: Customer buys multiple items (e.g., sunglasses with quantity 2).

Brainstorming

PRACTICAL CASE STUDY FOR FACT TABLES
E-COMMERCE
DATA COLLECTION POINTS

- **Other Data Points**:
- **Picking Date**
- **Order Fulfillment**
- **Supplier Delivery**
- **Additional Data**: Gathered across various stages of order processing and logistics in the warehouse.

PRACTICAL CASE STUDY FOR FACT TABLES E-COMMERCE
PRACTICAL STEPS TO DESIGN FACT TABLES

Brainstorming

1. **Identify Business Process**: Analyzing e-commerce sales, order processing, and logistics.
2. **Define Grain**: Decide granularity level (e.g., one row per order line item).
3. **Identify Dimensions**:
    1. **Time Dimensions**: Picking date, order fulfillment date.
    2. **Product Dimensions**: Product category, type, etc.
    3. **Customer Dimensions**: Customer name, customer ID.
4. **Identify Facts**:
    1. Sales amount
    2. Quantity
    3. Unit price
    4. Other operational metrics related to order processing and logistics.

Brainstorming

PRACTICAL CASE STUDY FOR FACT TABLES E-COMMERCE
APPLICATION OF STEPS

- **Data Integration**: Mapping natural keys (e.g., product IDs) to surrogate keys for consistency and efficiency.
- **Fact Table Design**: Ensuring each fact table aligns with defined grain and includes relevant dimensions for comprehensive data analysis.
- **Practical Example**: Creating fact tables for sales transactions, order processing metrics, and logistics operations based on gathered data points.

PRACTICAL CASE STUDY FOR FACT TABLES
E-COMMERCE
DESIGNING THE FACT TABLE

lentify Business Process:

1. Analyzing e-commerce operations, focusing on sales, promotions, logistics, and profitability.

efine Grain:

1. Granularity decisions to capture detailed transactional data for comprehensive analysis.

lentify Dimensions:

1. Time Dimensions: Order date, fulfillment date.
2. Product Dimensions: Product category, type.
3. Customer Dimensions: Customer demographics, purchasing behavior.

4. **Identify Facts**:
   1. Sales Amount
   2. Quantity Sold
   3. Profit Margins
   4. Discount Rates
   5. Operational Efficiency Metrics (related to logistics)

Brainstorming

Brainstorming

PRACTICAL CASE STUDY FOR FACT TABLES E-COMMERCE
MANAGERIAL GOALS AND INSIGHTS

- **Logistics Efficiency**: Focus on optimizing warehouse operations.
- **Profit Maximization**: Overall goal, considering factors like profit margins, promotions, and discounts.
  - **Impact of Discounts**: Potential to increase sales volume but also risks associated with profit margins.
  - **Complex Analysis**: Requires strategic analysis to balance profitability and sales volume.

Brainstorming

PRACTICAL CASE STUDY FOR FACT TABLES E-COMMERCE
STRATEGIC APPROACH

- **Data Warehouse Creation**: Essential for strategic analysis and decision-making.
- **Fact Table Design**: Aligning with business goals to enhance operational efficiency and profitability.
- **Next Steps**: Detailed exploration in the upcoming lecture focusing on business process identification.

MANDATORY STEPS

PRACTICAL CASE STUDY FOR FACT TABLES
E-COMMERCE
STEP 1

dentify Relevant Business Process:

- Basis of data warehouse design
- Crucial for company operations
- Consider data availability and quality

electing Business Process:

- Sales transactions chosen
- Critical for understanding product sales, profitability, and performance
- Analyze across categories and websites

nalytical Goals:

- Determine products sold and profitability
- Evaluate sales and profit performance
- Analyze across various dimensions (e.g., categories, websites)
- Track performance trends over time

lext Steps:

Sales transactions

Which products sold?
What is sales profit?
Sales of each website?
Performance on different days?
Sales over time?

PRACTICAL CASE STUDY FOR FACT TABLES E-COMMERCE STEP II

- **Declaring the Grain:**
  - Determine the level of detail for each row in the fact table
  - Optimal analytical value with atomic grain (highest level of detail)
  - Example: Each row represents one item order line in a customer's order
  - Ensures highest dimensionality and analytical capability
- **Transactional Fact Table Example:**
  - Each row represents a single item in a customer's shopping cart
  - Provides detailed insight into individual transactions
  - Avoids data aggregation which may reduce dimensionality and analytical depth
- **Benefits of Atomic Grain:**
  - Maximizes analytical value by retaining detailed dimensions
  - Generally manageable performance with modern database capabilities
  - Supports a wide range of analytical use cases
- **Next Steps:**
  - Proceed to define dimensions for comprehensive data analysis

PRACTICAL CASE STUDY FOR FACT TABLES E-COMMERCE STEP II

name_order_id	no_order_times	totalquantity	sales_amount	datetime
1001	1	1	900	2023-06-14 08:30:00
1002	2	3	1080	2023-06-14 09:15:00
1003	1	1	380	2023-06-14 10:00:00

customer_id	customer_name	order_id	order_line_name	order_line_id	quantity	unit_price	discounted price	promo_id	sales_amount	product_cost	datetime
1	Max Mustermann	1001	Laptop	5001	1	1000	900	101	900	800	2023-06-14 08:30:00
2	Erika Musterfrau	1002	Smartphone	5002	2	600	540	102	1080	900	2023-06-14 09:15:00
3	Hans Beispiel	1003	Tablet	5003	1	400	380	103	380	350	2023-06-14 10:00:00

PRACTICAL CASE STUDY FOR FACT TABLES E-COMMERCE STEP III

- **Finding Dimensions:**
  - Describe measures and context derived from defined grain
  - Dimensions include: customers, products (order lines), promotions, date/time features
  - Also identify system origin (websites) as a dimension
- **Derivation from Grain:**
  - Grain of order lines and orders defines associated dimensions naturally
  - Example: Each order line connects to customer, product, promotion, and timestamp dimensions
- **Implementation of Keys:**
  - Introduce primary key for unique row identification
    - Numeric integer assigned sequentially
  - Establish foreign keys for website and datetime dimensions
    - Maintain existing IDs in integer format
    - Rename for clarity as foreign keys

PRACTICAL CASE STUDY FOR FACT TABLES E-COMMERCE STEP 3

- **Datetime Dimension Handling:**
  - Consolidate datetime into a single dimension table
    - Convert datetime into integer format for uniformity and efficiency
    - Option to split into separate date and time dimensions, but opted for unified approach
- **Summary of Identified Dimensions:**
  - Customers, products (order lines), promotions, datetime, websites
- **Next Steps:**
  - Proceed to identify facts in the upcoming lecture

PRACTICAL CASE STUDY FOR FACT TABLES
E-COMMERCE
STEP 3

K	Website_FK	Customer_FK	Order_id	Order_line_FK	Quantity	Unit_price	Discounted_pi	promo_FK	sales_amc	product_cost	DataT
	101	201	301	401	2	25.00	20.00	501	40.00	15.00	601
	102	202	302	402	1	50.00	45.00	502	45.00	30.00	602
	103	203	303	403	3	15.00	12.00	503	36.00	9.00	603

- Customer
- Products
- Promotions
- Time/Date
- Website

PRACTICAL CASE STUDY FOR FACT TABLES E-COMMERCE STEP 4

- **Identifying Facts in Fact Table Design:**
  - Determine relevant measurements aligned with the defined grain of the fact table.
  - Ensure metrics comply with the granularity established; reconsider grain if necessary.
- **Compliance with Grain:**
  - Metrics like sales amount directly correspond to the row's transactional detail.
  - Verify all measurements fit the intended level of detail without aggregation issues.
- **Adjusting Facts:**
  - Consider removing unnecessary facts or introducing additional ones.
  - Derive new calculations from existing facts to enhance analytical capabilities.

PRACTICAL CASE STUDY FOR FACT TABLES
E-COMMERCE
STEP 4

- **Considerations for Additivity:**
  - Emphasize the importance of additivity in selected facts.
  - Ensure metrics like sales amount and product costs are additive across dimensions.

- **Example Metrics and Derivatives:**
  - **Sales Amount:** Directly reflects transactional sales for each row.
  - **Unit Price:** Useful despite non-additive nature; caution required in aggregate calculations.
  - **Discounted Price:** Consider non-additivity and inform users for proper calculation usage.
  - **Derived Fact - Absolute Discount:** Calculated from unit price and discount, aiding in straightforward add

- **Final Steps:**
  - Validate all chosen facts for additivity and analytical utility.
  - Confirm readiness to proceed to the next phase of fact table implementation.

WHICH FACTS ARE IN THE FACT TABLE

K	Website_FK	Customer_FK	Order_id	Order_line_FK	Quantity	Unit_price	Discounted_pi	promo_FK	sales_amc	product_cost	DataT
	101	201	301	401	2	25.00	20.00	501	40.00	15.00	601
	102	202	302	402	1	50.00	45.00	502	45.00	30.00	602
	103	203	303	403	3	15.00	12.00	503	36.00	9.00	603

PRACTICAL CASE STUDY FOR FACT TABLES E-COMMERCE STEP 4

- **Handling Percentage Discounts:**
  - Avoid storing percentage discounts due to their non-additive nature.
  - Recommend calculating these values dynamically in BI tools like Power BI or Tableau.
  - BI tools handle percentages and ratios effectively without storing them in the Data Warehouse physically.
- **Calculating Profit:**
  - Calculate profit by subtracting product cost from sales amount.
  - Ensure accuracy by multiplying product cost with quantity before subtraction.
  - Store profit physically in the Data Warehouse to maintain consistency and avoid discrepancies in calculatio

PRACTICAL CASE STUDY FOR FACT TABLES E-COMMERCE STEP 4

- **Advantages of Storing Profit:**
  - Mitigates risks associated with varied calculation methods and user errors.
  - Ensures uniformity in profit calculations across different users and analytical processes.

- **Final Fact Table Design:**
  - **Dimensions:** Website, Customer, Product (associated with order line), Date.
  - **Facts:** Quantity, Unit Price, Discounts, Sales Amount, Product Cost, Profit.
  - Comprehensive layout ensuring all necessary dimensions and facts are included for thorough analysis.

- **Next Steps:**
- Proceed to explore dimensions in greater detail in the upcoming section.
- Enhance understanding and utilization of dimensions within the context of the designed fact tabl

PRACTICAL CASE STUDY FOR FACT TABLES
E-COMMERCE
STEP 4

K	Website_FK	Customer_FK	Order_id	Order_line_FK	Quantity	Unit_price	Discounted_pi	promo_FK	sales_amc	product_cost	DataT
	101	201	301	401	2	25.00	20.00	501	40.00	15.00	601
	102	202	302	402	1	50.00	45.00	502	45.00	30.00	602
	103	203	303	403	3	15.00	12.00	503	36.00	9.00	603

Additive
Discount absolute yes/no?
Discount percentage yes/no?
Profit

RESULT

Fact Fact Fact Fact Fact

K	Website_FK	Customer_FK	Order_id	Order_line_FK	Quantity	Unit_price	Discounted_pi	promo_FK	sales_amc	product_cost	DataT
	101	201	301	401	2	25.00	20.00	501	40.00	15.00	601
	102	202	302	402	1	50.00	45.00	502	45.00	30.00	602
	103	203	303	403	3	15.00	12.00	503	36.00	9.00	603

Website

Customer

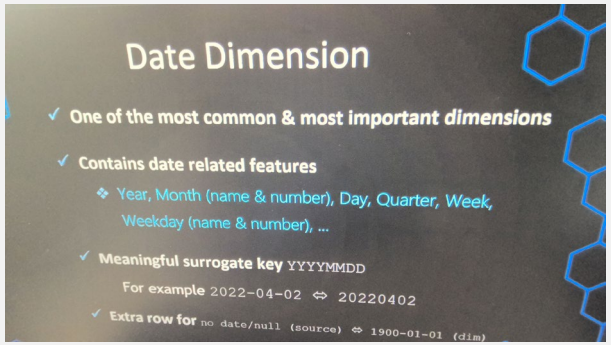
Products

DateTime

Dimensions

Additive
Discount absolute yes/no?
Discount percentage yes/no?

Fact

EXAM TOPICS

- 30 - 45 Questions
  - Chapter I - Term definitions
  - Chapter II – Architecture, Layers
  - Chapter III – Dimensional Modelling
  - Chapter IV – Facts
  - Chapter V – Dimensions
  - Chapter VI – ETL PRocess