[639] Data Modeling Approaches

Meenakshi Syamkumar

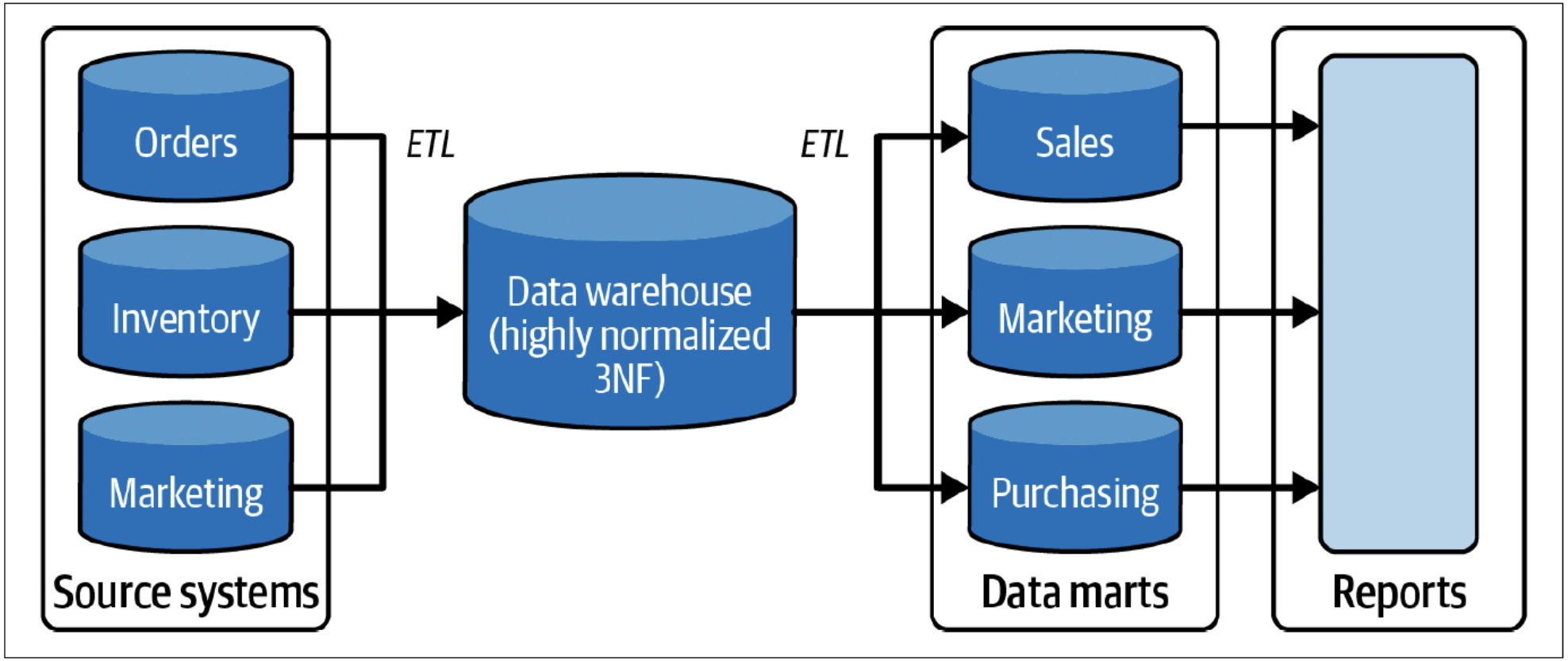
Learning Objectives

Understand	batch data modeling techniques like Inmon, Kimball, and Data Vaults
Understand	the two types of distributed joins

Batch Data Modeling: Inmon data model

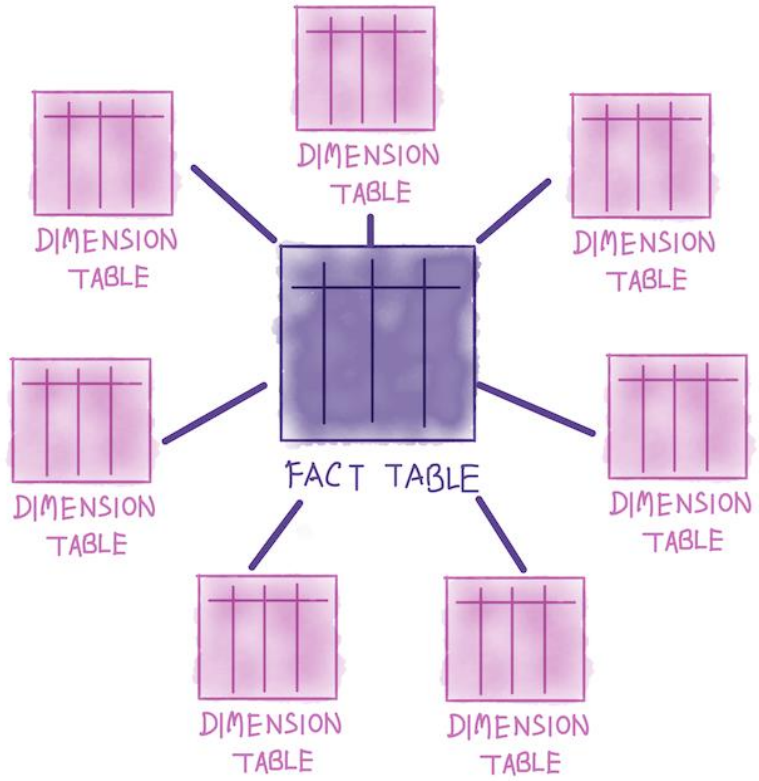
- Building a centralized, integrated, and normalized (3NF) data warehouse that serves as “a single source of truth” for the entire organization
- Top-down approach: Starts with a normalized data warehouse and then builds data marts as needed
- Four critical parts of a data warehouse:
  - Subject-oriented: The data warehouse focuses on a specific subject area, such as sales or marketing.
  - Integrated: Data from disparate sources is consolidated and normalized.
  - Nonvolatile: Data remains unchanged after data is stored in a data warehouse.
  - Time-variant: Varying time ranges can be queried.

Inmon data model: example



Batch Data Modeling: Kimball data model

- Organize data into fact and dimension tables
- Bottom-up approach: model and serve department or business analytics in the data warehouse itself
- Star Schema
  - Fact tables: measurable quantitative data or events
  - Dimension tables: contextual information for the data in fact tables



Kimball data model: Fact tables

- Factual, quantitative, and event-related data
- Immutable, numerical data type only, append-only, narrow and long tables (few columns, many rows)
- Lowest granularity of data
- Aggregations or derivations happen in a downstream query, data mart table or view
- Fact tables only reference dimensions and not other fact tables

OrderID	CustomerKey	DateKey	GrossSalesAmt
100	5	20220301	100.00
101	7	20220301	75.00
102	7	20220301	50.00

Kimball data model: Dimension tables

- Reference data, attributes, and relational context for the events stored in fact tables
- Wide and short tables with denormalized data (duplication possible)
- Describe: what, where, and when.

DateKey	Date-ISO	Year	Quarter	Month	Day-of-week
20220301	2022-03-01	2022	1	3	Tuesday
20220302	2022-03-02	2022	1	3	Wednesday
20220303	2022-03-03	2022	1	3	Thursday

CustomerKey	FirstName	LastName	ZipCode	EFF_StartDate	EFF_EndDate
5	Joe	Reis	84108	2019-01-04	9999-01-01
7	Matt	Housley	84101	2020-05-04	2021-09-19
7	Matt	Housley	84123	2021-09-19	9999-01-01
11	Lana	Belle	90210	2022-02-04	9999-01-01

Kimball data model: Slowly Changing Dimensions (SCD)

- Type 1
  - Overwrite existing dimension records.
  - No access to the deleted historical dimension records.
- Type 2
  - Keep a full history of dimension records.
  - Recording changes: previous record is flagged as changed, and a new dimension record is created.
- Type 3
  - Record change creates a new field.

TOP HAT

Kimball data model: Slowly Changing Dimensions (SCD)

CustomerKey	FirstName	LastName	ZipCode
7	Matt	Housley	84101

CustomerKey	FirstName	LastName	Original ZipCode	Current ZipCode	CurrentDate
7	Matt	Housley	84101	84123	2021-09-19

- Type 1 is default in most data warehouses
- Type 2 is most commonly used

Kimball data mode: Star Schema

- Unlike highly normalized approaches to data modeling, the star schema is a fact table surrounded by the necessary dimensions.
- Advantages:
  - Results in fewer joins than other data models, which speeds up query performance.
  - Easier for business users to understand and use.
- Typically: multiple star schemas address different facts of the business
- Conformed dimension: dimension that is reused across multiple star schemas, thus sharing the same fields

Batch Data Modeling: Data Vault model

- Separates the structural aspects of a source system's data from its attributes
- Data Vault simply loads data from source systems directly into a handful of purpose-built tables in an insert-only manner
- Three types of tables:
  - Hub: stores business keys
  - Link: maintains relationships among business keys
  - Satellite: represents a business key's attributes and context
- A user will query a hub, which will link to a satellite table containing the query's relevant attributes

Data Vault model: Hubs

- The central entity of a Data Vault that retains a record of all unique business keys loaded into the Data Vault
- Insert-only, permanent table
- Standard fields:
  - Hash key: The primary key used to join data between systems. This is a calculated hash field (MD5 or similar).
  - Load date: The date the data was loaded into the hub.
  - Record source: The source from which the unique record was obtained.
  - Business key(s): The key used to identify a unique record.

OrderHashKey	LoadDate	RecordSource	OrderID
f899139df5...	2022-03-01	Website	100
38b3eff8ba...	2022-03-01	Website	101
ec8956637a...	2022-03-01	Website	102

ProductHashKey	LoadDate	RecordSource	ProductID
4041fd80ab...	2020-01-02	ERP	1
de8435530d...	2021-03-09	ERP	2
cf27369bd8...	2021-03-09	ERP	3

Data Vault model: Links

- Tracks the relationships of business keys between hubs
- Connect hubs, ideally at the lowest possible grain
- Many to many links:
  - create a new link that ties business concepts (or hubs) to represent the new relationship

OrderProductHashKey	LoadDate	RecordSource	ProductHashKey	OrderHashKey
ff64ec193d...	2022-03-01	Website	4041fd80ab...	f899139df5...
ff64ec193d...	2022-03-01	Website	de8435530d...	f899139df5...
e232628c25...	2022-03-01	Website	cf27369bd8...	38b3eff8ba...
26166a5871...	2022-03-01	Website	4041fd80ab...	ec8956637a...

Data Vault model: Satellites

- Descriptive attributes that give meaning and context to hubs
- Satellites can connect to either hubs or links
- Required fields:
  - a primary key consisting of the business key of the parent hub
  - a load date

ProductHashKey	LoadDate	RecordSource	ProductName	Price
4041fd80ab...	2020-01-02	ERP	Thingamajig	50
de8435530d...	2021-03-09	ERP	Whatchamacallit	25
cf27369bd8...	2021-03-09	ERP	Whozeewhatzit	75

Wide denormalized tables

- Wide table: a highly denormalized and very wide collection of many fields
- Typically stored in a columnar database
- Fields may contain single values or contain nested data
- Analytics queries on wide tables often run faster than equivalent queries on highly normalized data requiring many joins
- Use case scenarios:
  - Recommendation systems, advertisement attribution, event logs and telemetry systems
  - Technologies: Google BigTable, Apache Cassandra, Snowflake, BigQuery, Elasticsearch

Data Transformation

Transformation

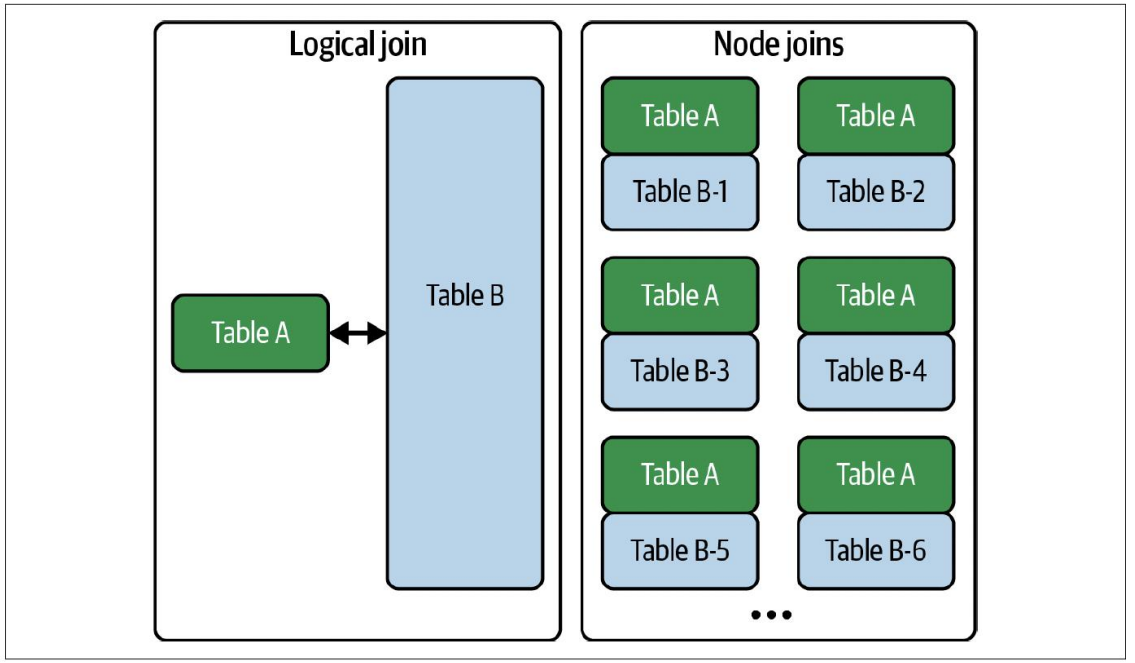
- Transformations manipulate, enhance, and save data for downstream use, increasing its value in a scalable, reliable, and cost-effective manner.
- Transformation versus querying:
  - A query retrieves the data from various sources based on filtering and join logic
  - A transformation persists the results for consumption by additional transformations or queries
- Build complex pipelines that combine data from multiple sources and reuse intermediate results for multiple final outputs
- Transformation pipelines span not only multiple tables and datasets but also multiple systems

Batch Transformations

- Operate on discrete chunks of data
- Typically operate on a fixed schedule (e.g., daily, hourly, or every 15 minutes)
- Distributed joins
  - Broadcast joins
  - Shuffle hash joins

Broadcast joins

- Asymmetric join:
  - one large table distributed across nodes
  - one small table that can easily fit on a single node
- Query engine “broadcasts” the small table (table A) out to all nodes, where it gets joined to the parts of the large table (table B)
- Prefiltering data to create broadcast joins where possible can dramatically improve performance and reduce resource consumption



Shuffle hash joins

- Neither table is small enough to fit into a single node
- Storage partitioning will typically have no relation to the join key
- Hashing scheme to repartition data
  - Data is reshuffled to the appropriate node
  - Need to pull related data from both tables to the same place
- More resource intensive than broadcast joins