Copyright Notice

DeepLearning.AI

Data Transformation, Modeling and Serving

Data Modeling I

DeepLearning.AI

Intro to Data Modeling for Analytics

Welcome to Course 4

DeepLearning.AI

Data Modeling, Transformation, and Serving

Course 4 Overview

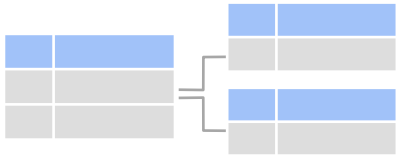
What is a Data Model?

Data Model

A data model organizes and standardizes data in a precise structured representation to enable and guide human and machine behavior, inform decision-making, and facilitate actions.

Define the structure, relationships and meaning of data.

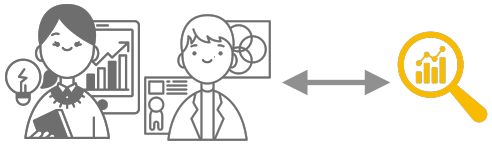
Modeling tabular data



- What tables make up the model?
- How the tables relate to one another?
- What columns to choose for each table?

Structure the data in a way that connects back to the organization

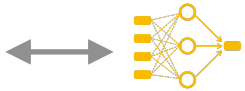
Data is understandable & valuable



Analytics

Data is meaningful

Machine



Machine Learning

Good Data Models

Reflect the business goals and logic while incorporating business rules

Ensure compliance with operational standards and legal requirements

Outline the relationships between business processes

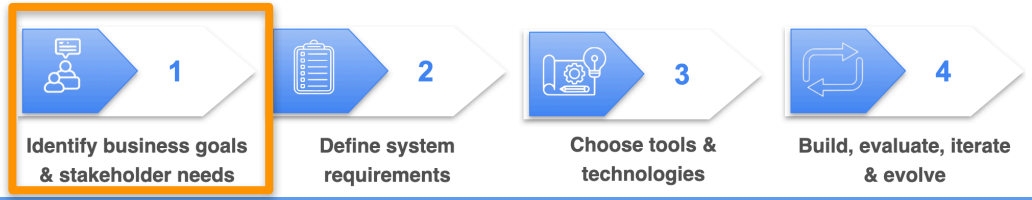
Serve as a powerful communication tool, creating a shared language

Example: *Define what constitutes an "active user"*

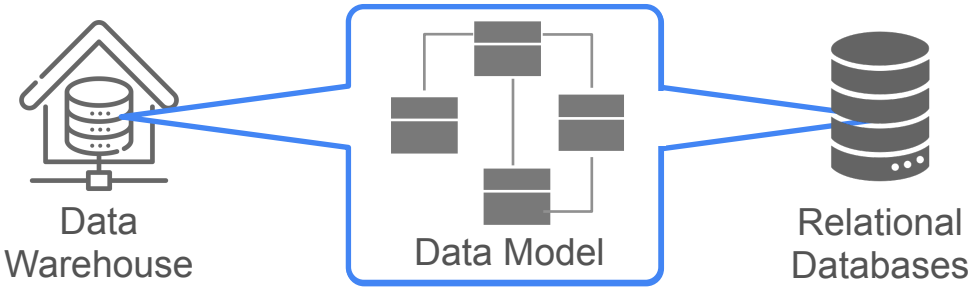
- *Logged into their account in the last 30 days*
- *Made a purchase in the previous six months*

Good Data Models	Poor Data Models
Reflect the business goals and logic while incorporating business rules	Don't reflect how the business operates
Ensure compliance with operational standards and legal requirements	Create more problems than they solve
Outline the relationships between business processes	Provide stakeholders with inaccurate information and create confusion
Serves as a powerful communication tool, creating a shared language	

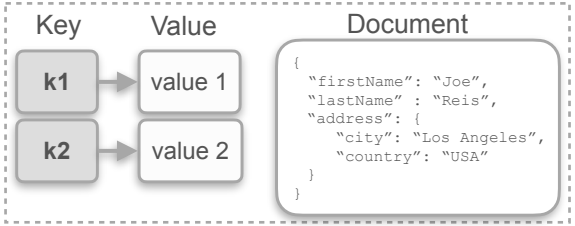
Ensure Successful Data Modeling



The Practice of Data Modeling



Data Lake 1.0



NoSQL Databases

Big Data
Systems

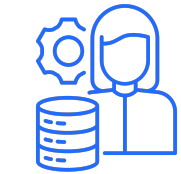
Data Swamps

Ignore traditional data modeling

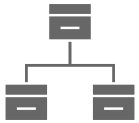
Data Modeling

Targeted Data Modeling Approach

Focus on specific business domains

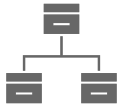


Data Engineer



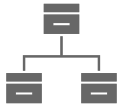
Marketing

Better understand customer behavior and campaign effectiveness



Finance

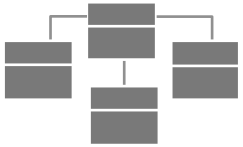
Analyze spending patterns and identify cost-saving opportunities



Machine Learning

Drive better decision-making and impactful AI models

Week 1 **Batch data modeling**

- The three levels of data modeling: **Conceptual**, **Logical** and **Physical**
- Two Basic schemas:  Normalized  Star Schema
- Popular modeling techniques for analytical use cases:
  - Inmon and Kimball modeling approaches
  - Data Vaults and One Big Table

Week 2 **Data modeling and transformation for machine learning**

Week 3 **Transformation deep-dive**

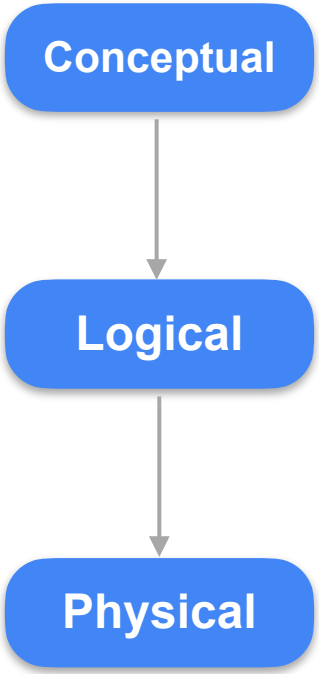
Week 4 **Build an end-to-end data pipeline**

DeepLearning.AI

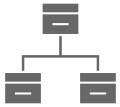
Intro to Data Modeling for Analytics

Conceptual, Logical, and Physical Data Models

Data Models



Conceptual



Describes business entities, relationships, & attributes

Business logic and rules

Entity-Relationship (ER) Diagram

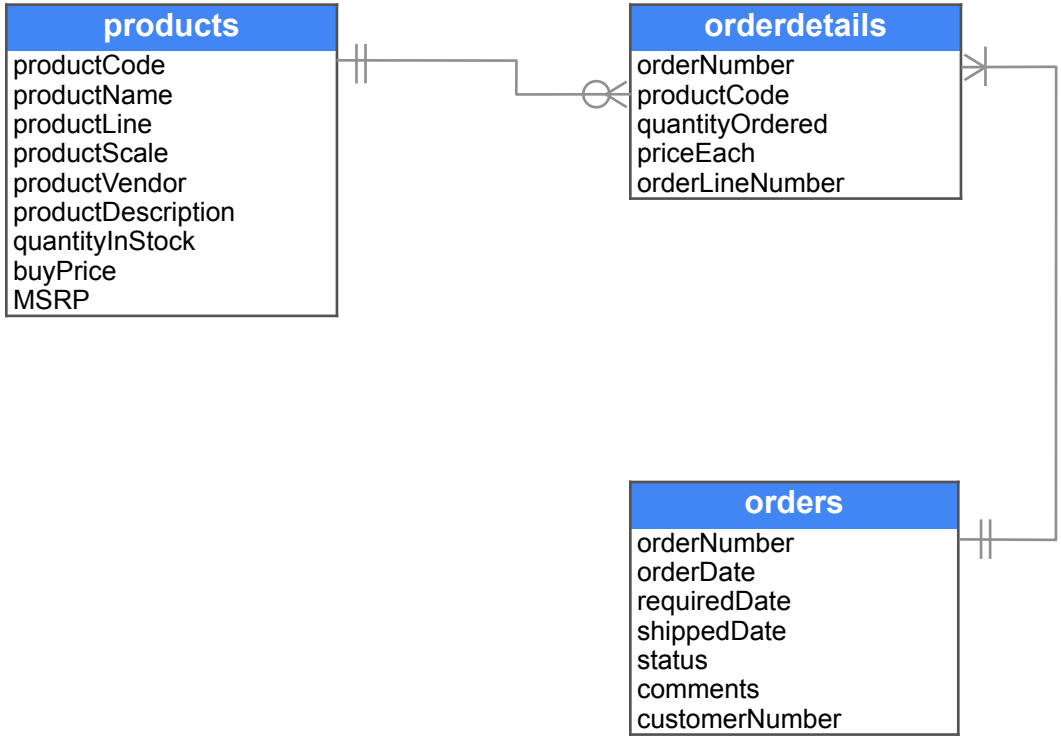
A standard tool for visualizing the relationships among various business entities.

Conceptual

Describes business entities, relationships, & attributes

Business logic and rules

Entity-Relationship (ER) Diagram



Conceptual

Describes business entities, relationships, & attributes

Business logic and rules

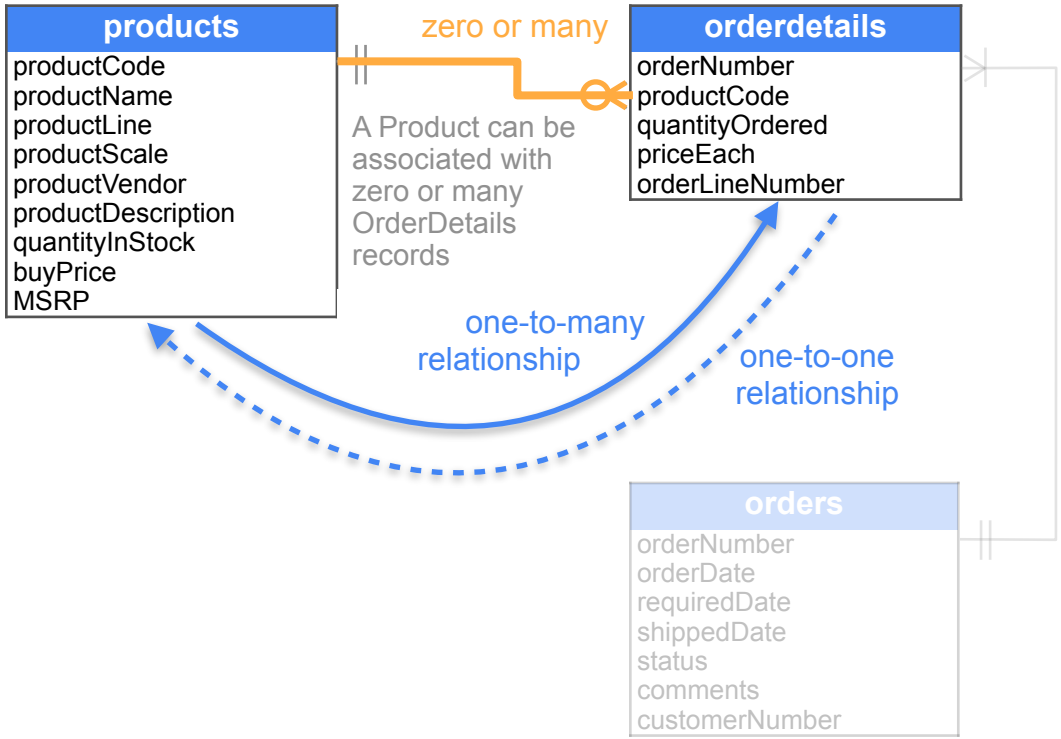
Entity-Relationship (ER) Diagram

Conceptual

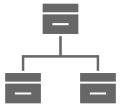
Describes business entities, relationships, & attributes

Business logic and rules

Entity-Relationship (ER) Diagram



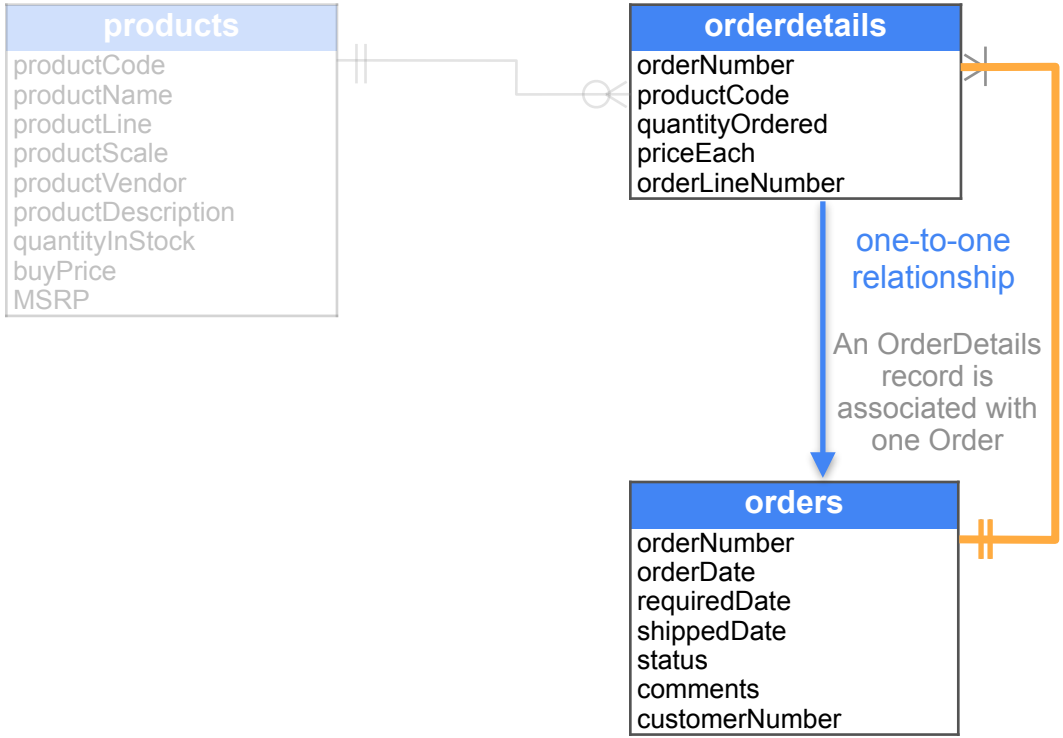
Conceptual



Describes business entities, relationships, & attributes

Business logic and rules

Entity-Relationship (ER) Diagram

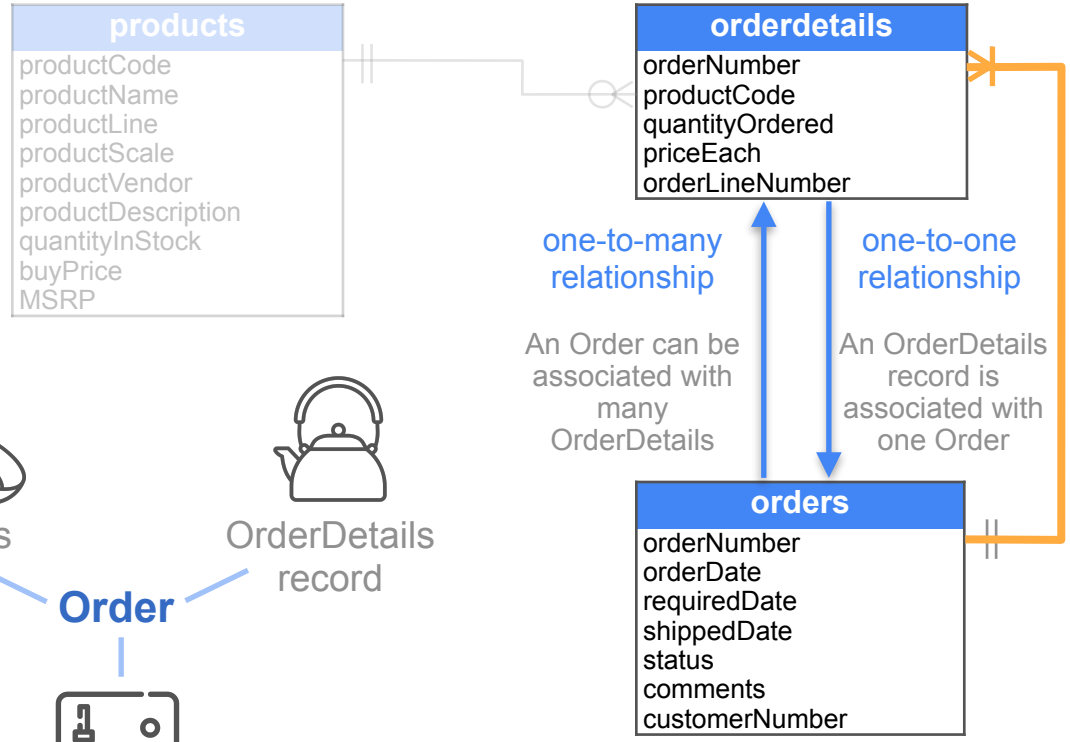


Conceptual

Describes business entities, relationships, & attributes

Business logic and rules

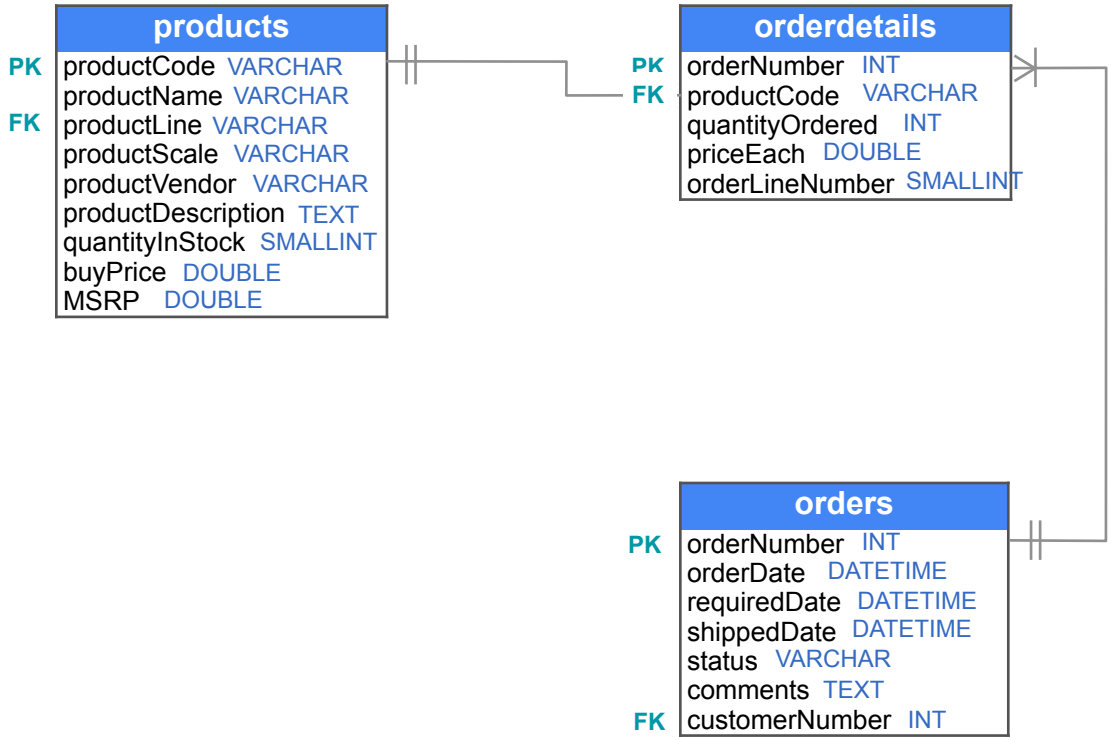
Entity-Relationship (ER) Diagram



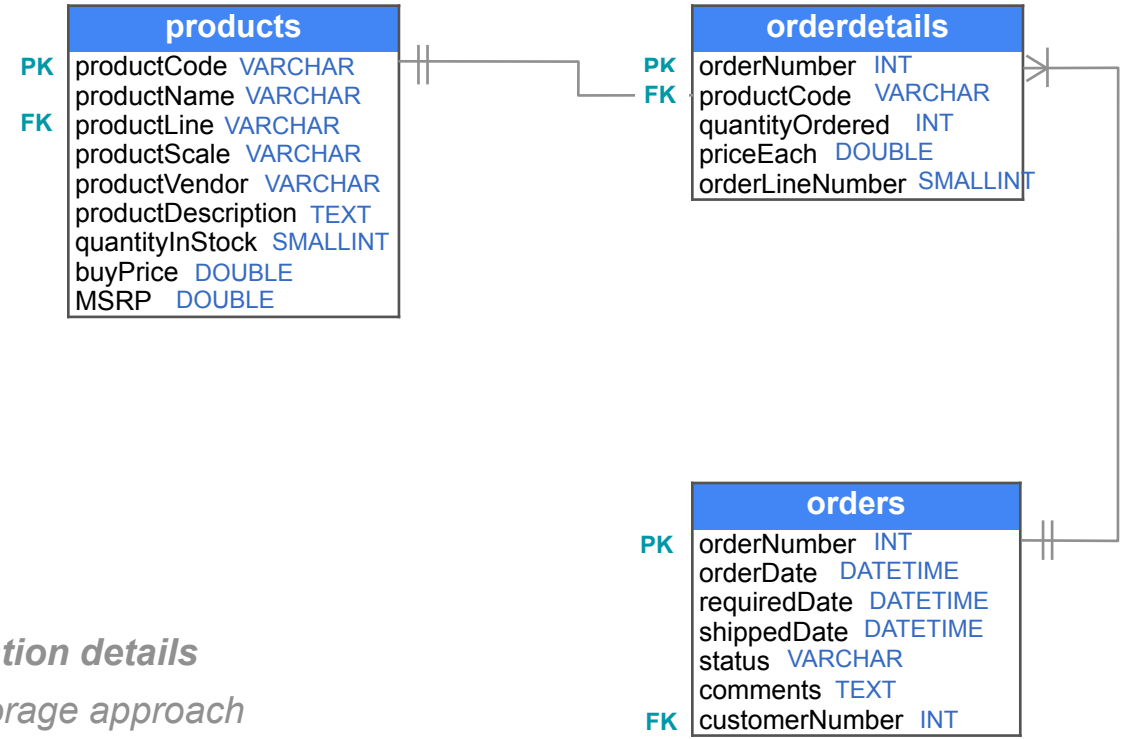
Logical

Details about the implementation of the conceptual model

Entity-Relationship (ER) Diagram



Entity-Relationship (ER) Diagram



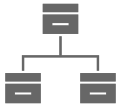
Physical

Details about the implementation of the logical model in a specific DBMS

Configuration details

- *Data storage approach*
- *Partitioning details*
- *Replication details*

Conceptual



Describes business entities, relationships, & attributes

Business logic and rules

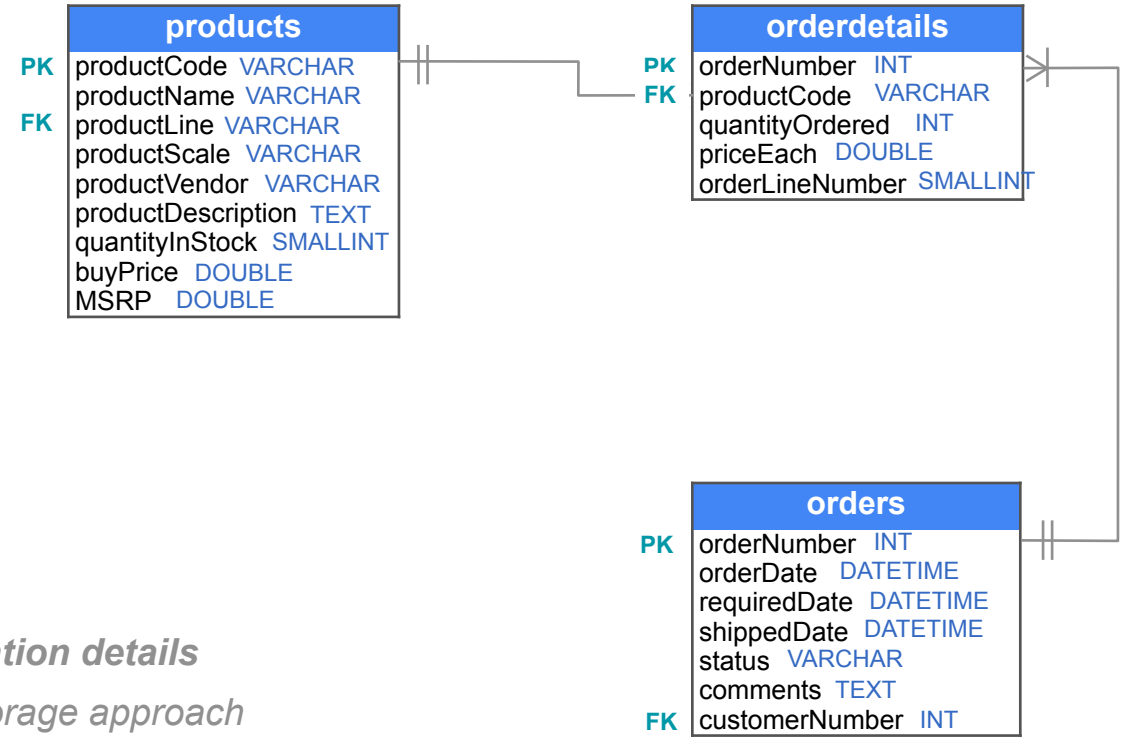
Logical

Details about the implementation of the conceptual model

Physical

Details about the implementation of the logical model in a specific DBMS

Entity-Relationship (ER) Diagram



Configuration details

- *Data storage approach*
- *Partitioning details*
- *Replication details*

DeepLearning.AI

Intro to Data Modeling for Analytics

Normalization

Normalization

Normalization

A data modeling practice typically applied to relational databases to remove the redundancy of data within a database and ensure referential integrity between tables.

Edgar Codd

Codd's Objectives of Normalization:

- To free the collection of relations from undesirable insertion, update, and deletion dependencies
- To reduce the need for restructuring the collection of relations, as new types of data are introduced

SalesOrders

OrderID	ItemNumber	sku	price	quantity	name	CustomerID	CustomerName	address	OrderDate
100	1	1	50	1	Thingamajig	5	Joe Reis	1st. St	1/08/2024
100	2	2	25	2	Whatchamacallit	5	Joe Reis	1st. St	1/08/2024
101	1	3	75	1	Whoozeewhatzit	7	Matt Housely	2nd Ave.	1/08/2024
101	2	2	25	3	Whatchamacallit	7	Matt Housely	2nd Ave.	1/08/2024
102	1	1	50	1	Thingamjig	9	Colleen Fotsch	3rd Lk	1/08/2024

Customers

CustomerID	CustomerName	address
5	Joe Reis	1st. St
7	Matt Housely	2nd Ave.
9	Colleen Fotsch	3rd Lk

Items

sku	price	name
1	50	Thingamajig
2	25	Whatchamacallit
3	75	Whoozeewhatzit

Order Items

OrderID	ItemNumber	sku	quantity
100	1	1	1
100	2	2	2
101	1	3	1
101	2	2	3
102	1	1	1

Orders

OrderID	CustomerID	Date
100	5	1/08/2024
101	7	1/08/2024
102	9	1/08/2024

SalesOrders

OrderID	ItemNumber	sku	price	quantity	name	CustomerID	CustomerName	address	OrderDate
100	1	1	50	1	Thingamajig	5	Joe Reis	1st. St	1/08/2024
100	2	2	25	2	Whatchamacallit	5	Joe Reis	1st. St	1/08/2024
101	1	3	75	1	Whoozeewhatzit	7	Matt Housely	2nd Ave.	1/08/2024
101	2	2	25	3	Whatchamacallit	7	Matt Housely	2nd Ave.	1/08/2024
102	1	1	50	1	Thingamjig	9	Colleen Fotsch	3rd Lk	1/08/2024

Customers

CustomerID	CustomerName	address
5	Joe Reis	1st. St
7	Matt Housely	2nd Ave.
9	Colleen Fotsch	3rd Lk

Items

sku	price	name
1	50	Thingamajig
2	25	Whatchamacallit
3	75	Whoozeewhatzit

Less normalized:
Contains more redundant data

Order Items

OrderID	ItemNumber	sku	quantity
100	1	1	1
100	2	2	2
101	1	3	1
101	2	2	3
102	1	1	1

Orders

OrderID	CustomerID	Date
100	5	1/08/2024
101	7	1/08/2024
102	9	1/08/2024

SalesOrders

OrderID	ItemNumber	sku	price	quantity	name	CustomerID	CustomerName	address	OrderDate
100	1	1	50	1	Thingamajig	5	Joe Reis	1st. St	1/08/2024
100	2	2	25	2	Whatchamacallit	5	Joe Reis	1st. St	1/08/2024
101	1	3	75	1	Whoozeewhatzit	7	Matt Housely	2nd Ave.	1/08/2024
101	2	2	25	3	Whatchamacallit	7	Matt Housely	2nd Ave.	1/08/2024
102	1	1	50	1	Thingamjig	9	Colleen Fotsch	3rd Lk	1/08/2024

Customers

CustomerID	CustomerName	address
5	Joe Reis	1st. St
7	Matt Housely	2nd Ave.
9	Colleen Fotsch	3rd Lk

Items

sku	price	name
1	50	Thingamajig
2	25	Whatchamacallit
3	75	Whoozeewhatzit

Order Items

OrderID	ItemNumber	sku	quantity
100	1	1	1
100	2	2	2
101	1	3	1
101	2	2	3
102	1	1	1

Orders

OrderID	CustomerID	Date
100	5	1/08/2024
101	7	1/08/2024
102	9	1/08/2024

SalesOrders

OrderID	ItemNumber	sku	price	quantity	name	CustomerID	CustomerName	address	
100	1	1	50	1	Thingamajig	5	Joe Reis	1st. St	
100	2	2	25	2	Whatchamacallit	5	Joe Reis	1st. St	1/08/2024
101	1	3	75	1	Whoozeewhatzit	7	Matt Housely	2nd Ave.	1/08/2024
101	2	2	25	3	Whatchamacallit	7	Matt Housely	2nd Ave.	1/08/2024
102	1	1	50	1	Thingamjig	9	Colleen Fotsch	3rd Lk	1/08/2024

First Normal Form

Customers

CustomerID	CustomerName	address
5	Joe Reis	1st. St
7	Matt Housely	2nd Ave.
9	Colleen Fotsch	3rd Lk

Items

sku	price	name
1	50	Thingamajig
2	25	Whatchamacallit
3	75	Whoozeewhatzit

Shipments

ShipmentID	OrderID	Ship.Details

Order Items

OrderID	ItemNumber	sku	quantity
100	1	1	1
100	2	2	2
101	1	3	1
101	2	2	3
102	1	1	1

Orders

OrderID	CustomerID	Date
100	5	1/08/2024
101	7	1/08/2024
102	9	1/08/2024

Third Normal Form

Denormalized Form

- contains redundant data
- contains nested data

1NF

- Each column must:
  - be unique
  - have a single value
- Unique primary key

SalesOrders

OrderID	OrderItems	CustomerID	CustomerName	address	OrderDate

100	[{
	“sku”:1,
	“price”:50,
	“quantity”: 1,
	“name”: “Thingamajig”
	}, {
	“sku”:2,
	“price”:25,
	“quantity”: 3,
	“name”: “Whatchamacallit”
	}]
		5	Joe Reis	1st. St	1/08/2024

101	[{
	“sku”:3,
	“price”:75,
	“quantity”: 1,
	“name”: “Whoozeewhatzit”
	}, {
	“sku”:2,
	“price”:25,
	“quantity”: 3,
	“name”: “Whatchamacallit”
	}]
		7	Matt Housely	2nd Ave.	1/08/2024

102	[{ “sku”:1, “price”:50, “quantity”: 1, “name”: “Thingamajig” }]	9	Colleen Fotsch	2nd Ave.	1/08/2024

1NF

SalesOrders

OrderID	sku	price	quantity	name	CustomerID	CustomerName	address	OrderDate
100	1	50	1	Thingamajig	5	Joe Reis	1st. St	1/08/2024
100	2	25	2	Whatchamacallit	5	Joe Reis	1st. St	1/08/2024
101	3	75	1	Whoozeewhatzit	7	Matt Housely	2nd Ave.	1/08/2024
101	2	25	3	Whatchamacallit	7	Matt Housely	2nd Ave.	1/08/2024
102	1	50	1	Thingamjig	9	Colleen Fotsch	3rd Lk	1/08/2024

1NF

SalesOrders

OrderID	ItemNumber	sku	price	quantity	name	CustomerID	CustomerName	address	OrderDate
100	1	1	50	1	Thingamajig	5	Joe Reis	1st. St	1/08/2024
100	2	2	25	2	Whatchamacallit	5	Joe Reis	1st. St	1/08/2024
101	1	3	75	1	Whoozeewhatzit	7	Matt Housely	2nd Ave.	1/08/2024
101	2	2	25	3	Whatchamacallit	7	Matt Housely	2nd Ave.	1/08/2024
102	1	1	50	1	Thingamjig	9	Colleen Fotsch	3rd Lk	1/08/2024

2NF

- The requirements of 1NF must be met
- **Partial dependencies** should be removed

Partial Dependency

A subset of non-key columns that depend on some columns in the composite key

1NF

SalesOrders

OrderID	ItemNumber	sku	price	quantity	name	CustomerID	CustomerName	address	OrderDate
100	1	1	50	1	Thingamajig	5	Joe Reis	1st. St	1/08/2024
100	2	2	25	2	Whatchamacallit	5	Joe Reis	1st. St	1/08/2024
101	1	3	75	1	Whoozeewhatzit	7	Matt Housely	2nd Ave.	1/08/2024
101	2	2	25	3	Whatchamacallit	7	Matt Housely	2nd Ave.	1/08/2024
102	1	1	50	1	Thingamjig	9	Colleen Fotsch	3rd Lk	1/08/2024

2NF

Order Items

Orders

order ID
100
101
102

3NF

- The requirements of 2NF must be met
- **Transitive dependencies** should be removed

Transitive Dependency

A non-key column depends on another non-key column

2NF

Order Items

OrderID	ItemNumber	sku	price	quantity	name
100	1	1	50	1	Thingamajig
100	2	2	25	2	Whatchamacallit
101	1	3	75	1	Whoozeewhatzit
101	2	2	25	3	Whatchamacallit
102	1	1	50	1	Thingamjig

Orders

order ID	CustomerID	CustomerName	address	OrderDate
100	5	Joe Reis	1st. St	1/08/2024
101	7	Matt Housely	2nd Ave.	1/08/2024
102	9	Colleen Fotsch	3rd Lk	1/08/2024

3NF

Order Items

OrderID	ItemNumber	sku	quantity
100	1	1	1
100	2	2	2
101	1	3	1
101	2	2	3
102	1	1	1

Orders

OrderID	CustomerID	OrderDate
100	5	1/08/2024
100	5	1/08/2024
101	7	1/08/2024
101	7	1/08/2024
102	9	1/08/2024

Customers

Items

2NF

Order Items

OrderID	ItemNumber	sku	price	quantity	name
100	1	1	50	1	Thingamajig
100	2	2	25	2	Whatchamacallit
101	1	3	75	1	Whoozeewhatzit
101	2	2	25	3	Whatchamacallit
102	1	1	50	1	Thingamjig

Orders

order ID	CustomerID	CustomerName	address	OrderDate
100	5	Joe Reis	1st. St	1/08/2024
101	7	Matt Housely	2nd Ave.	1/08/2024
102	9	Colleen Fotsch	3rd Lk	1/08/2024

3NF

Order Items

OrderID	ItemNumber	sku	quantity
100	1	1	1
100	2	2	2
101	1	3	1
101	2	2	3
102	1	1	1

Orders

OrderID	CustomerID	OrderDate
100	5	1/08/2024
100	5	1/08/2024
101	7	1/08/2024
101	7	1/08/2024
102	9	1/08/2024

Customers

CustomerID	CustomerName	address
5	Joe Reis	1st. St
7	Matt Housely	2nd Ave.
9	Colleen Fotsch	3rd Lk

Items

sku	price	name
1	50	Thingamajig
2	25	Whatchamacallit
3	75	Whoozeewhatzit

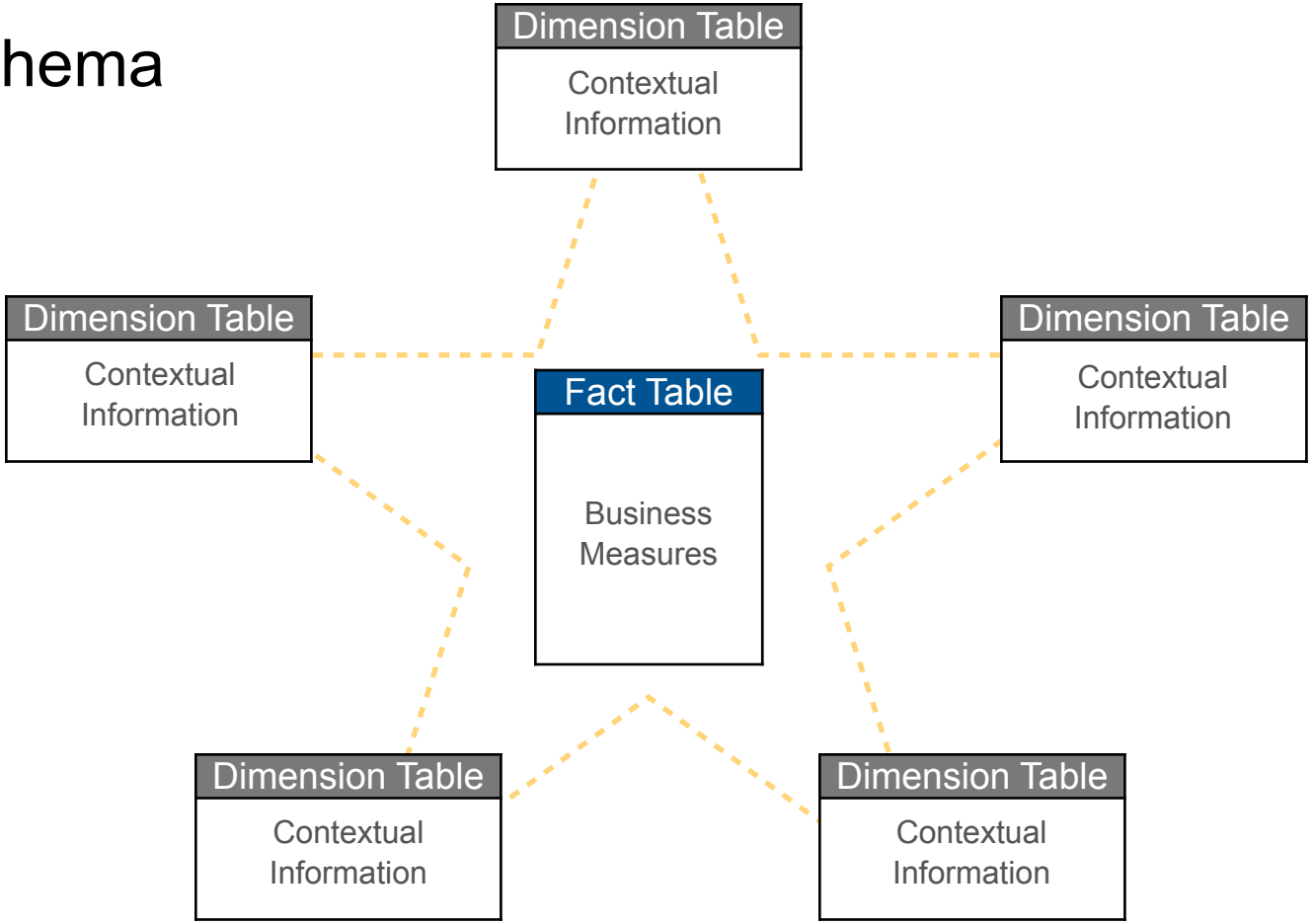
Convention: A normalized database means it's in third normal form.

DeepLearning.AI

Intro to Data Modeling for Analytics

Dimensional Modeling - Star Schema

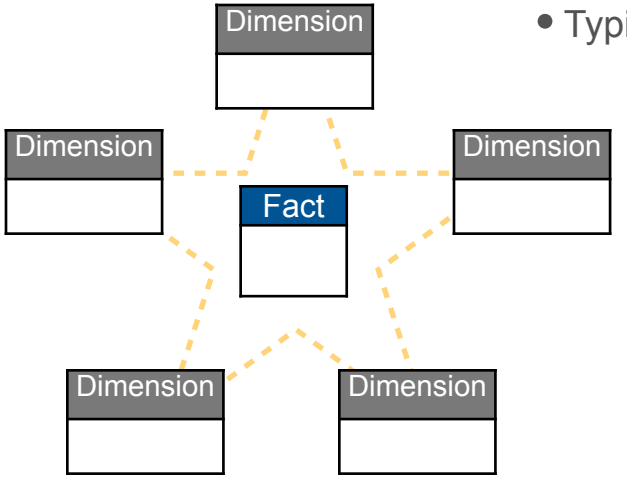
Star Schema


Dimension Table
Contextual Information
Dimension Table
Contextual Information
Fact Table
Business Measures
Dimension Table
Contextual Information
Dimension Table
Contextual Information
Dimension Table
Contextual Information

Fact Table

Contains quantitative business measurements that result from a business event or process.

- Each row contains the facts of a particular business event
- Immutable (append-only)
- Typically narrow and long

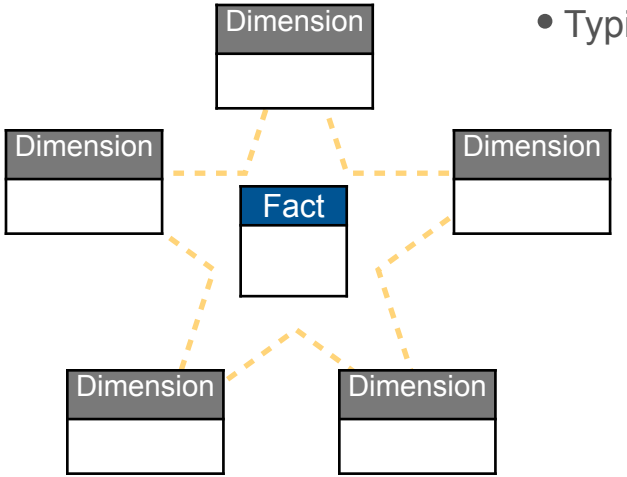


Business Event	Facts	Atomic Grain
Order a ride share	Trip duration, trip price, tip paid, trip delays, etc.	One completed ride by a customer

Fact Table

Contains quantitative business measurements that result from a business event or process.

- Each row contains the facts of a particular business event
- Immutable (append-only)
- Typically narrow and long



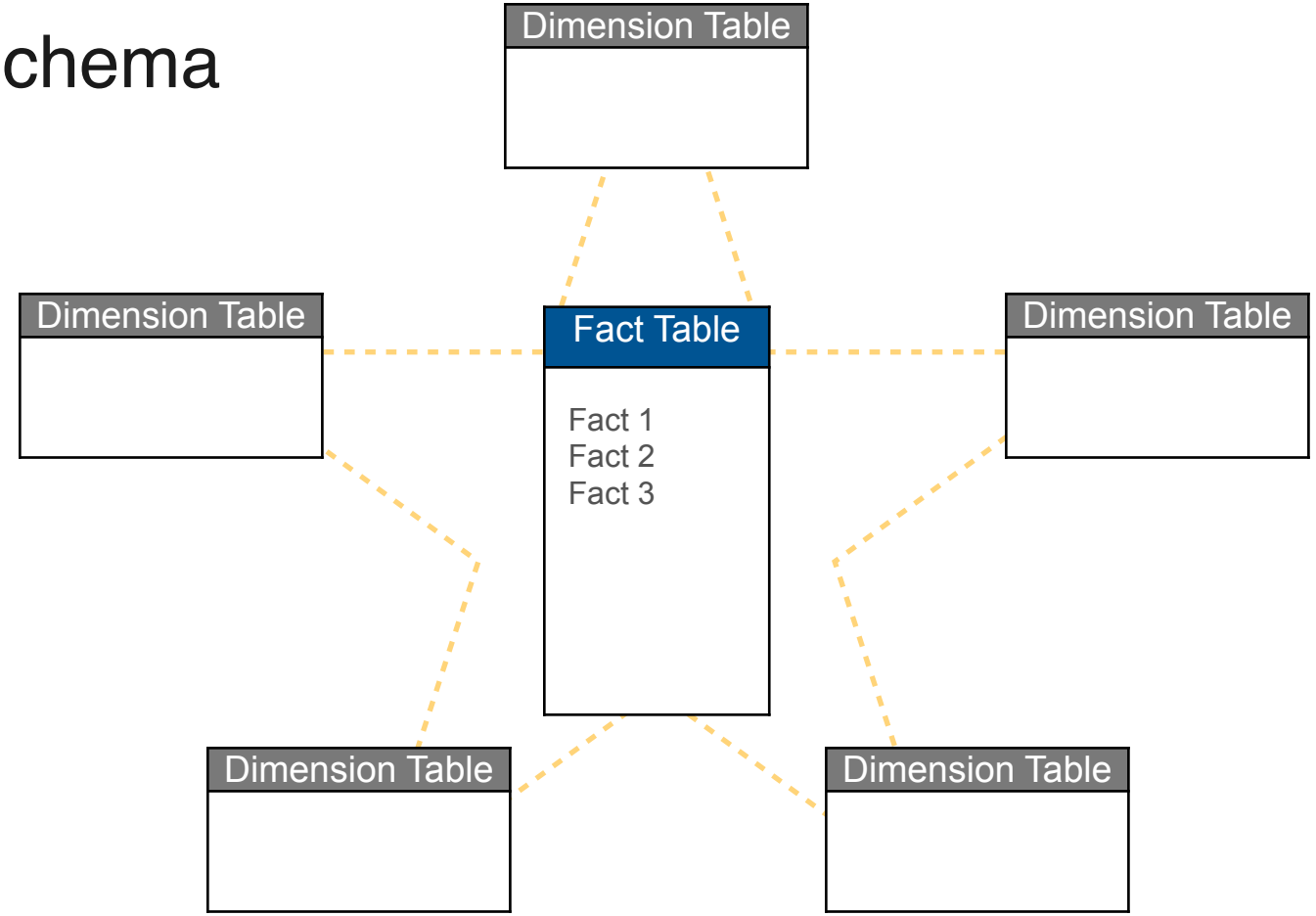
Business Event	Facts	Atomic Grain	Dimensions
Order a ride share	Trip duration, trip price, tip paid, trip delays, etc.	One completed ride by a customer	Customers, drivers, trip locations

Dimension Tables

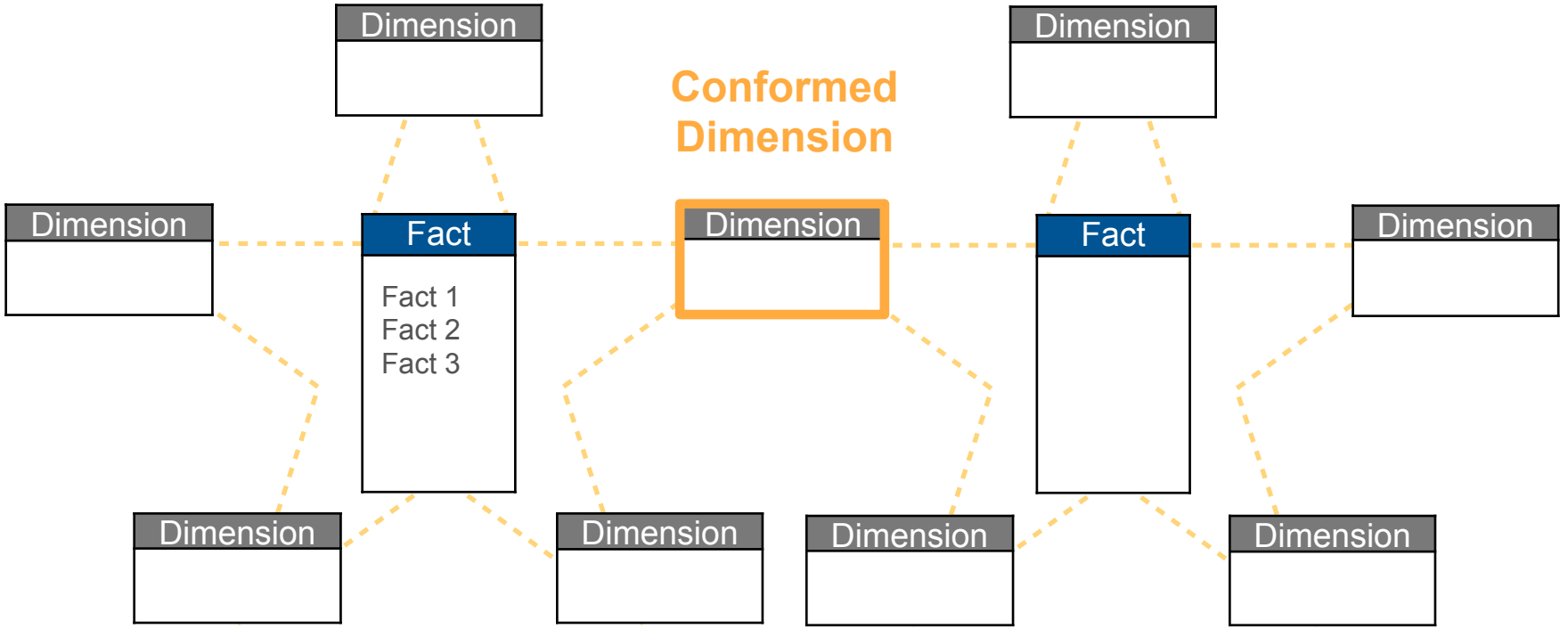
Provide the reference data, attributes and relational context for the events in the fact table.

- Describe the events' *what, who, where and when*
- Typically wide and short

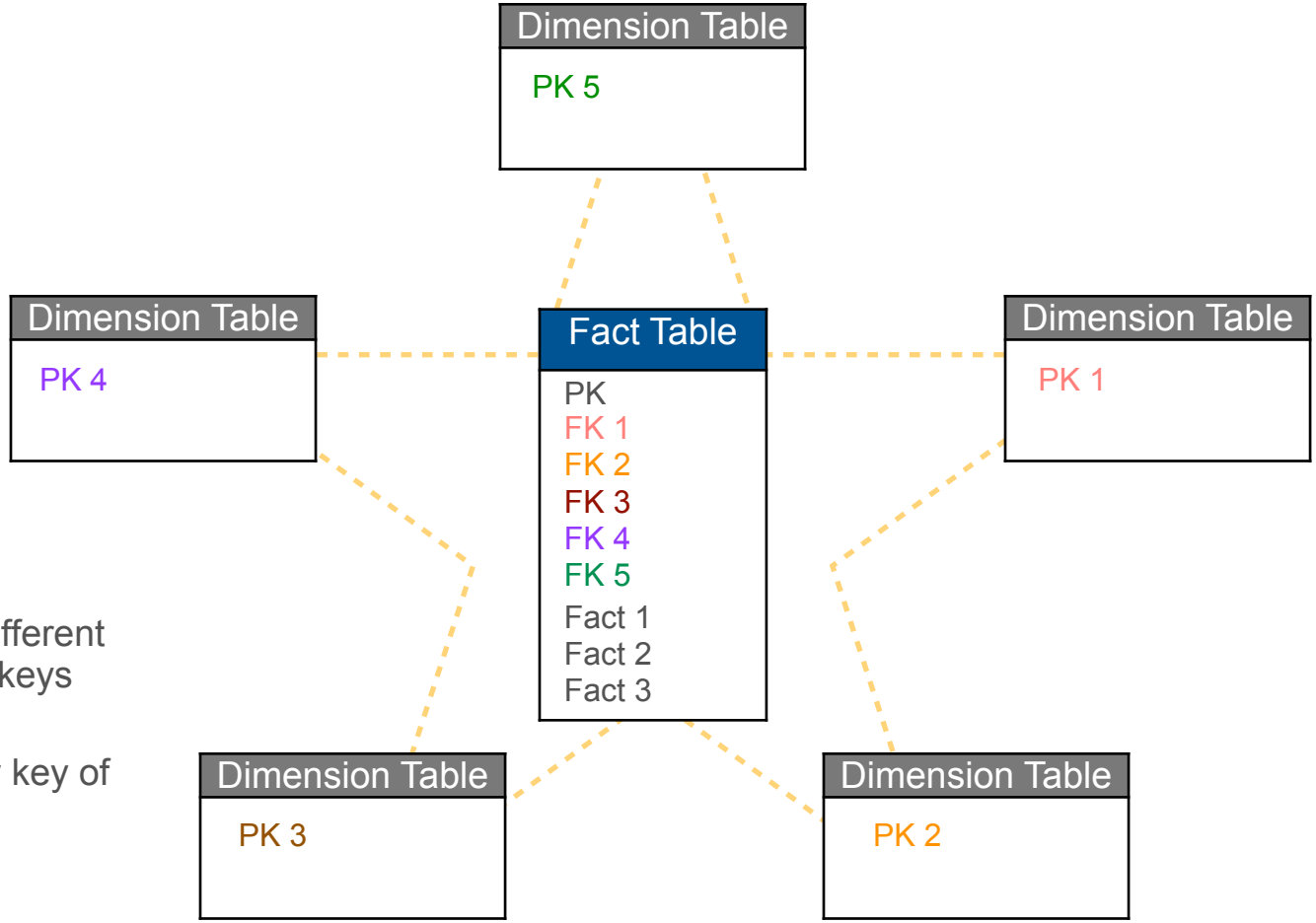
Star Schema


Dimension Table
Dimension Table
Fact Table
Fact 1
Fact 2
Fact 3
Dimension Table
Dimension Table
Dimension Table

Star Schema


Dimension
Dimension
Conformed Dimension
Dimension
Dimension
Dimension
Fact
Fact 1
Fact 2
Fact 3
Dimension
Fact
Dimension
Dimension
Dimension
Dimension
Dimension

Star Schema

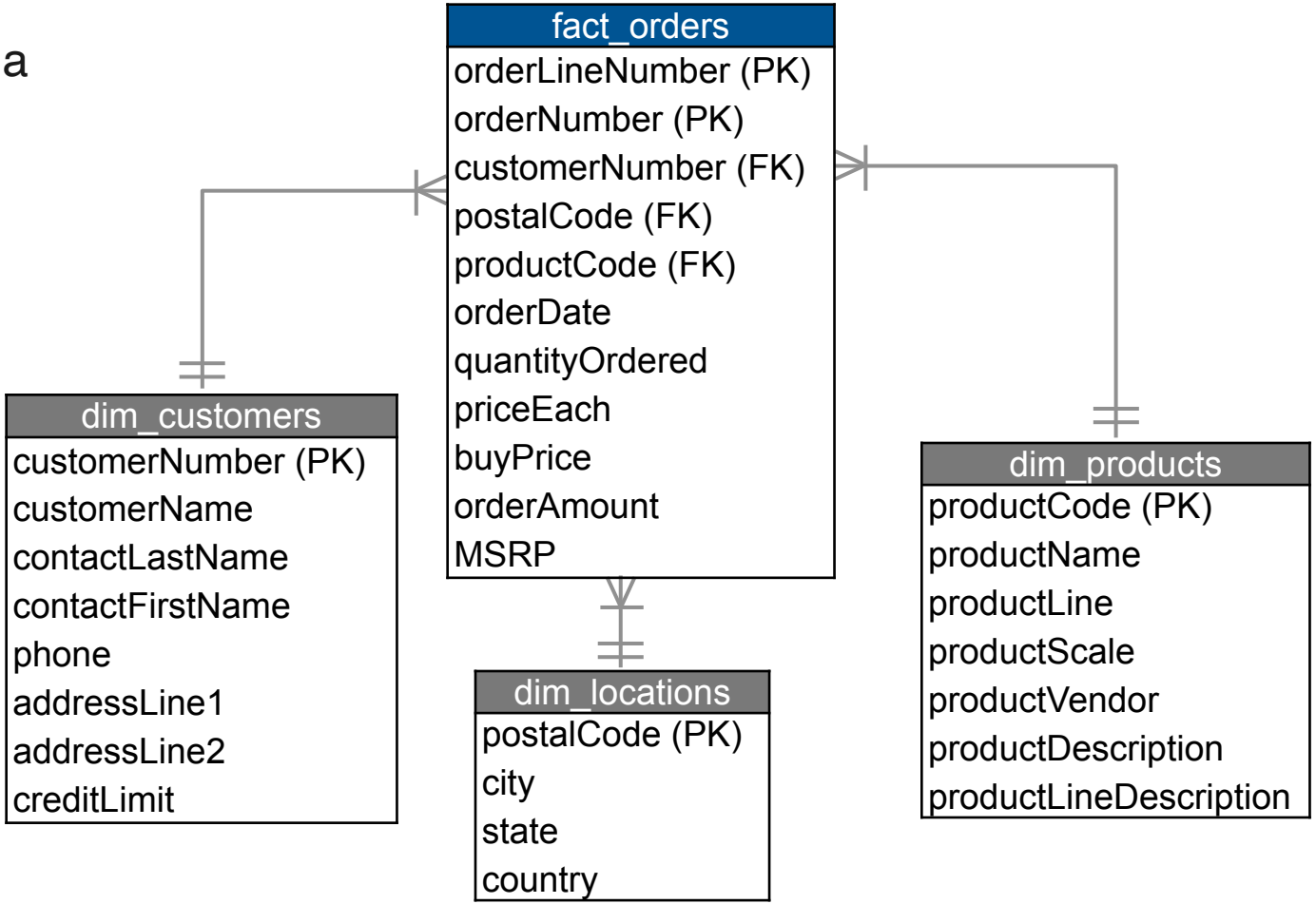


Best practice:

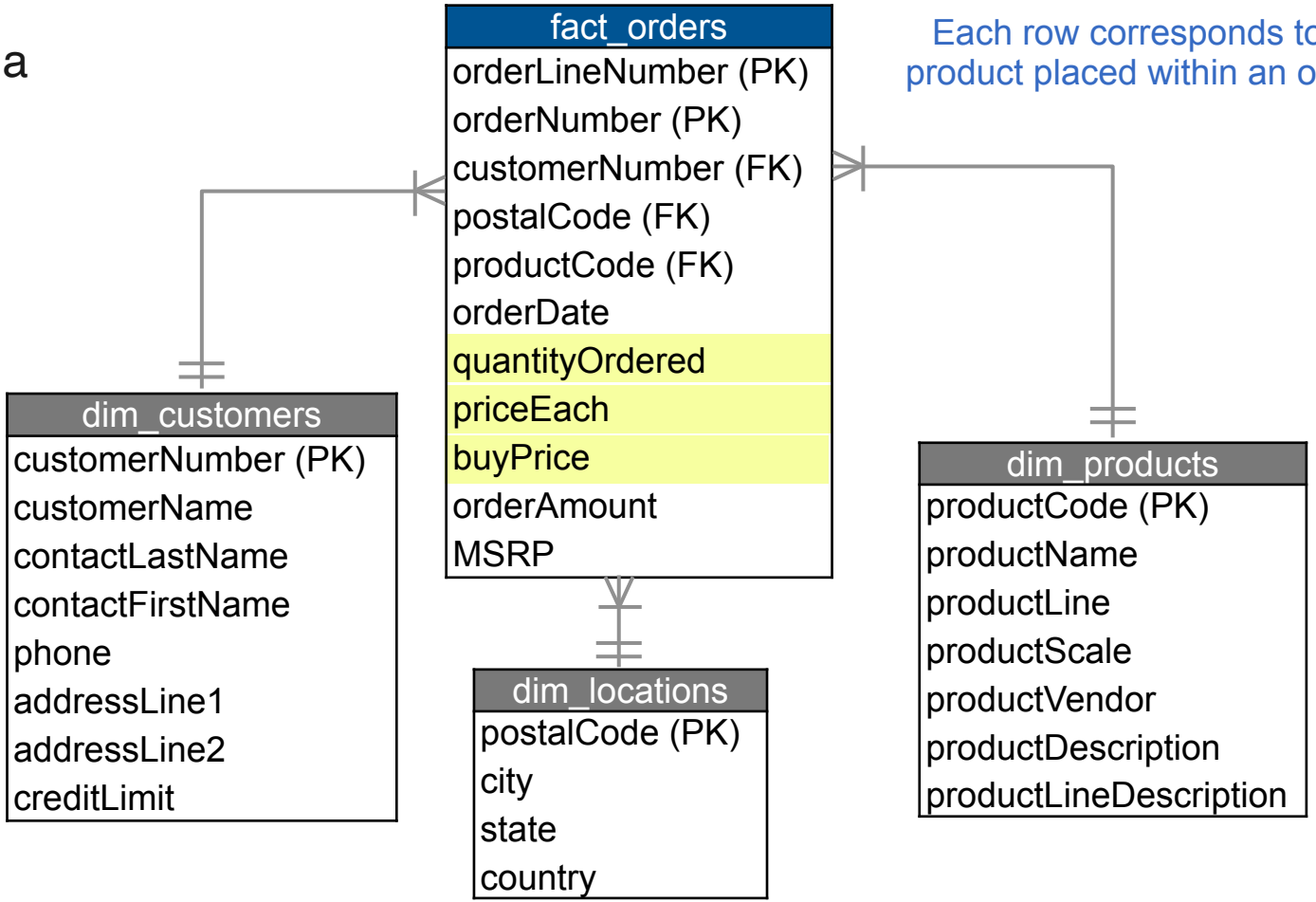
Create a surrogate key

- Used to combine data from different systems with natural primary keys that are in different formats
- Used to decouple the primary key of the star schema from source systems

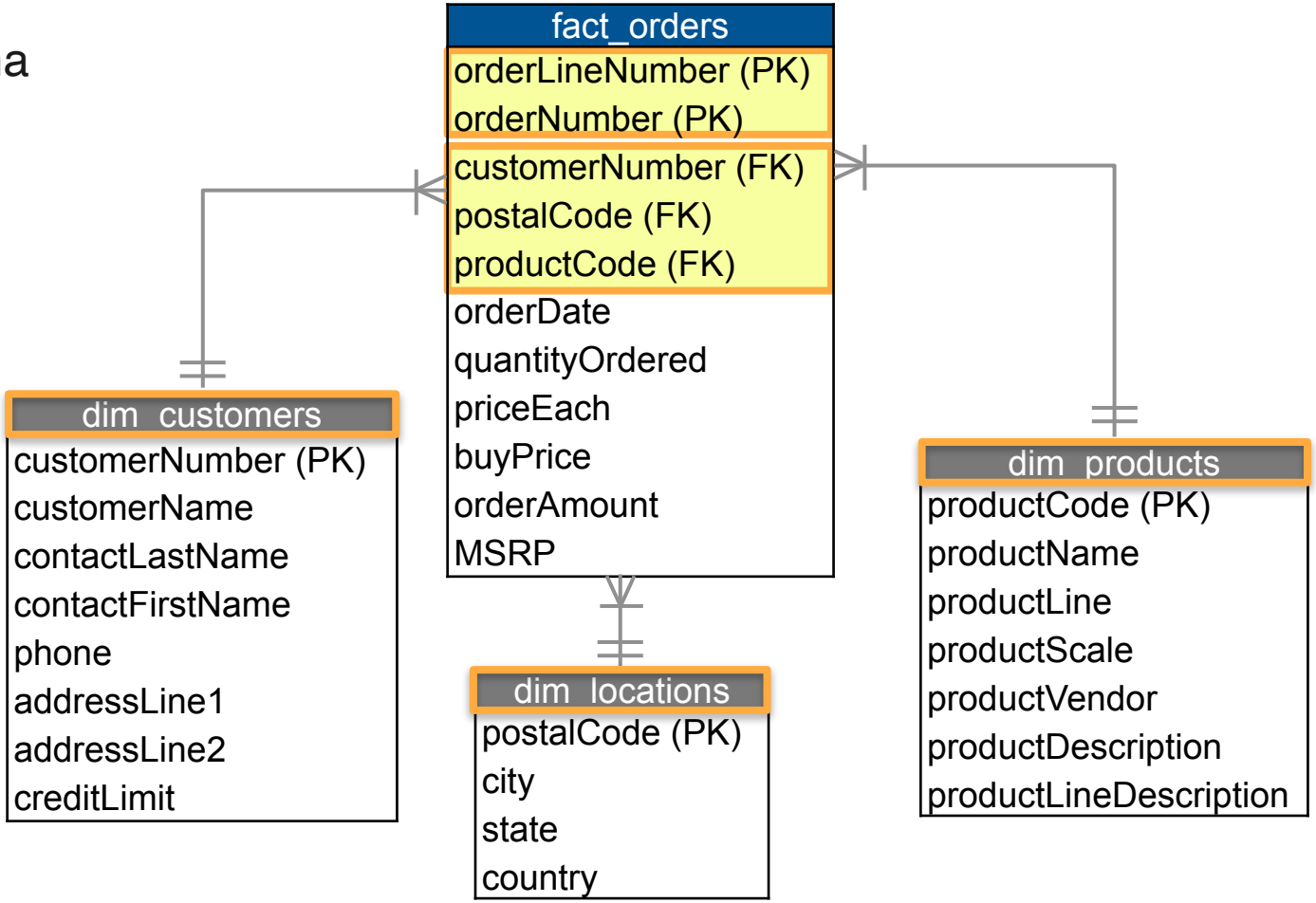
Star Schema Example



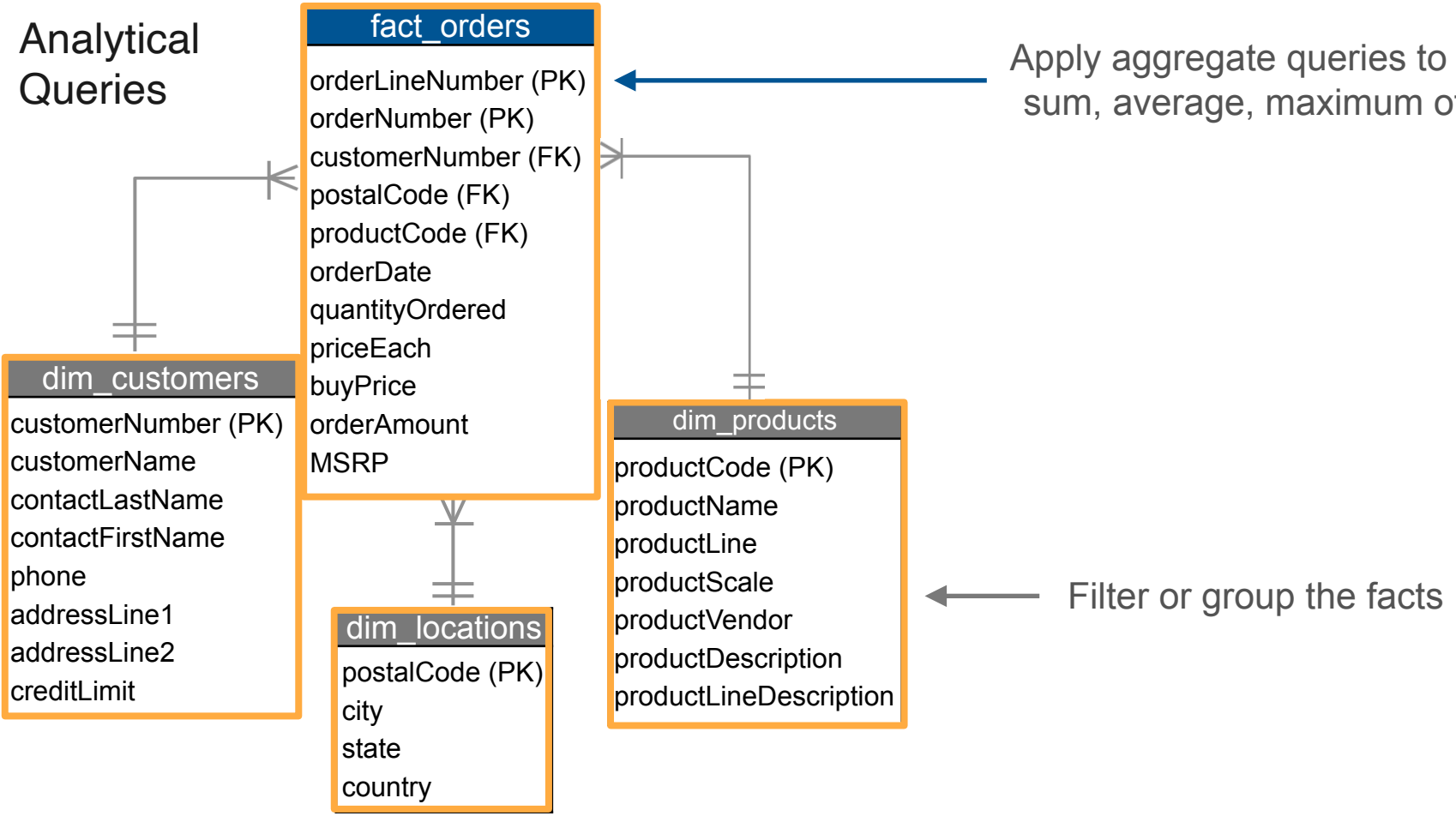
Star Schema Example



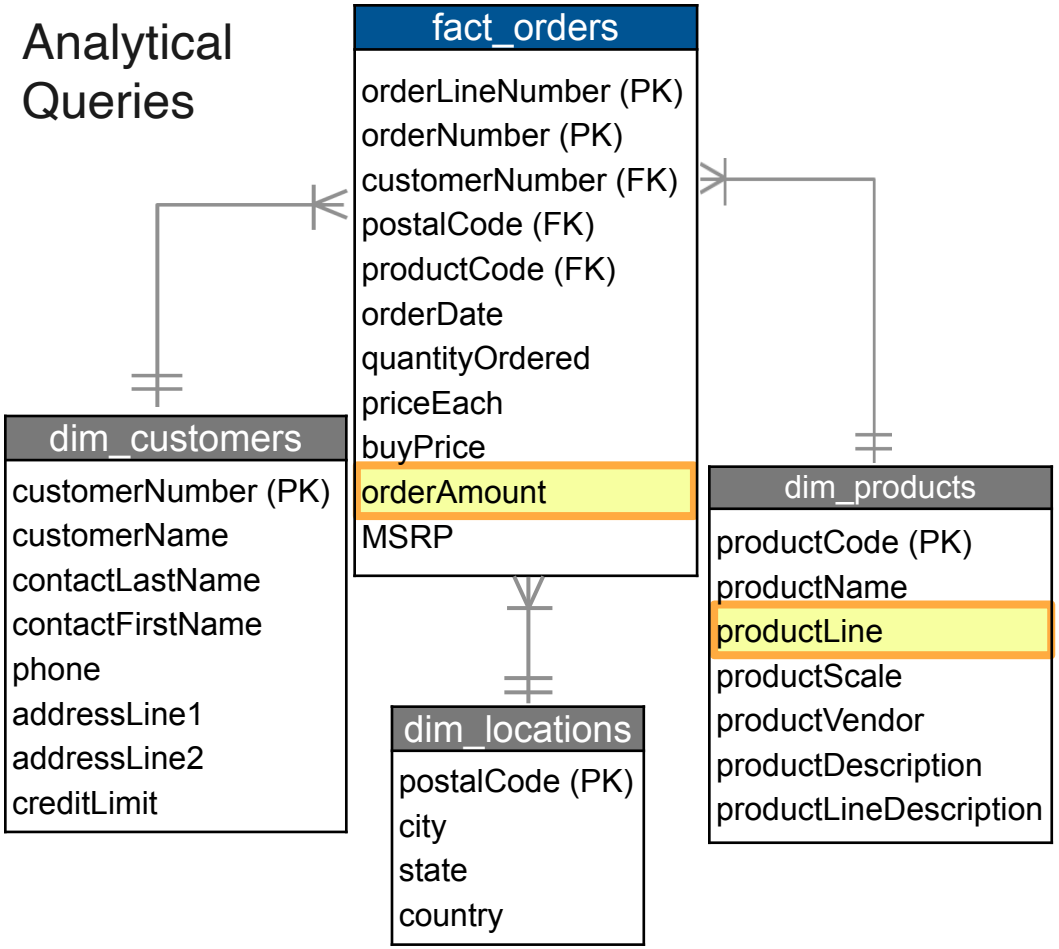
Star Schema Example



Analytical Queries



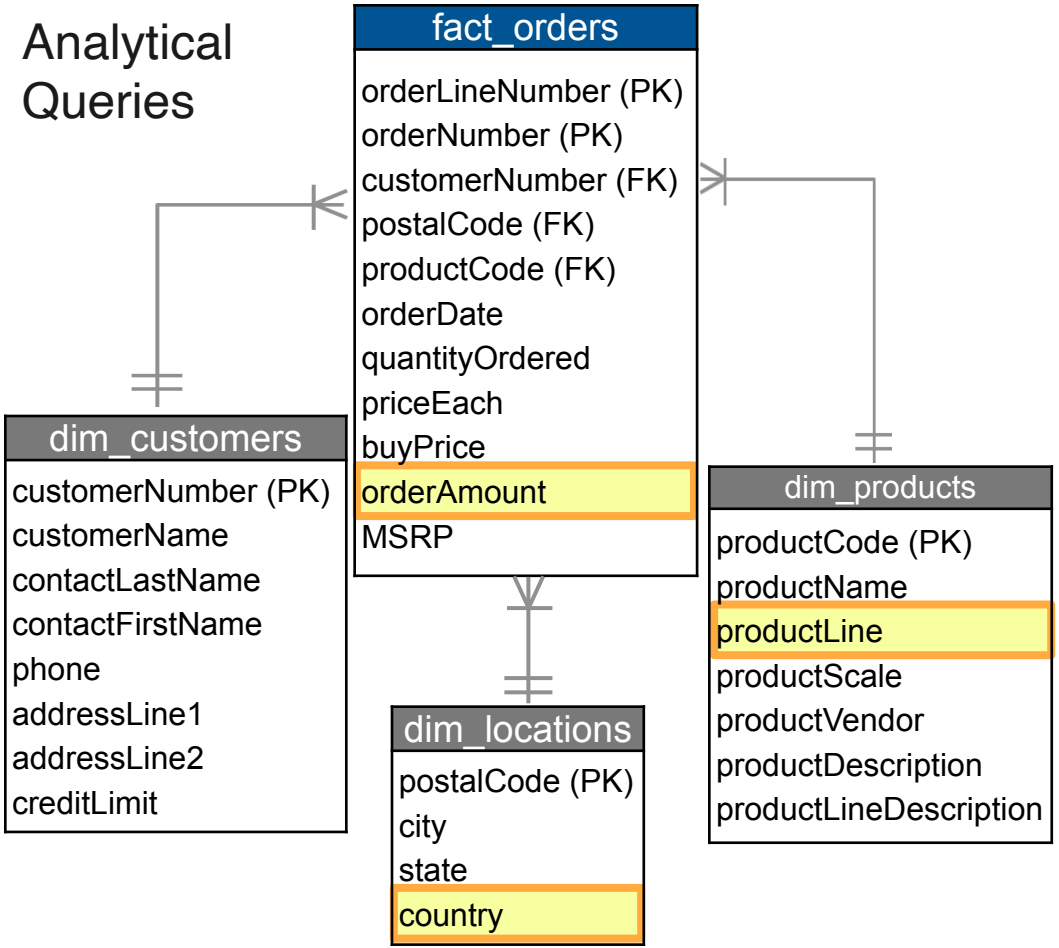
Analytical Queries



Find **the total sales amount** for each product line within the USA

```
SELECT
  SUM(fact_orders.orderAmount) AS total_sales
FROM fact_orders
JOIN dim_products ON
  fact_orders.productCode =
  dim_products.productCode
```

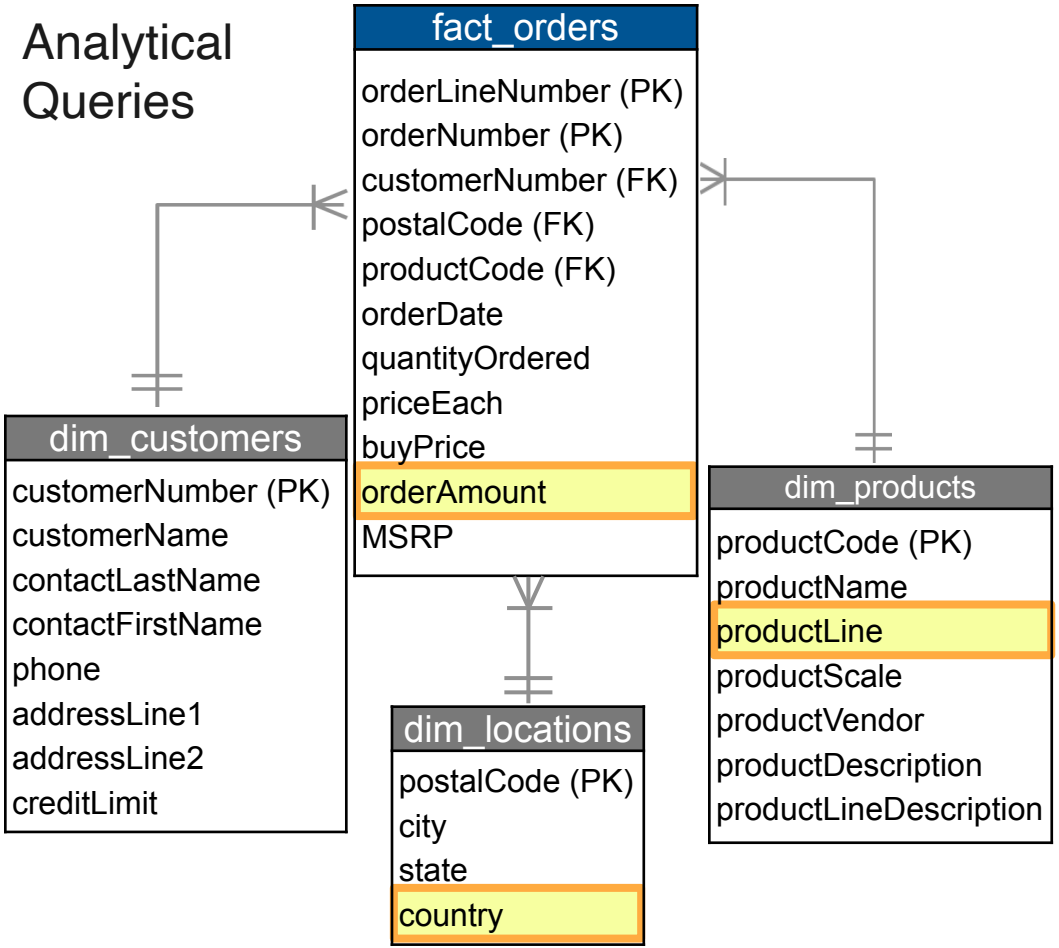
Analytical Queries



Find **the total sales amount** for each product line within the USA

```
SELECT
  dim_products.productLine,
  SUM(fact_orders.orderAmount) AS total_sales
FROM fact_orders
JOIN dim_products ON
  fact_orders.productCode =
  dim_products.productCode
GROUP BY dim_products.productLine
```

Analytical Queries

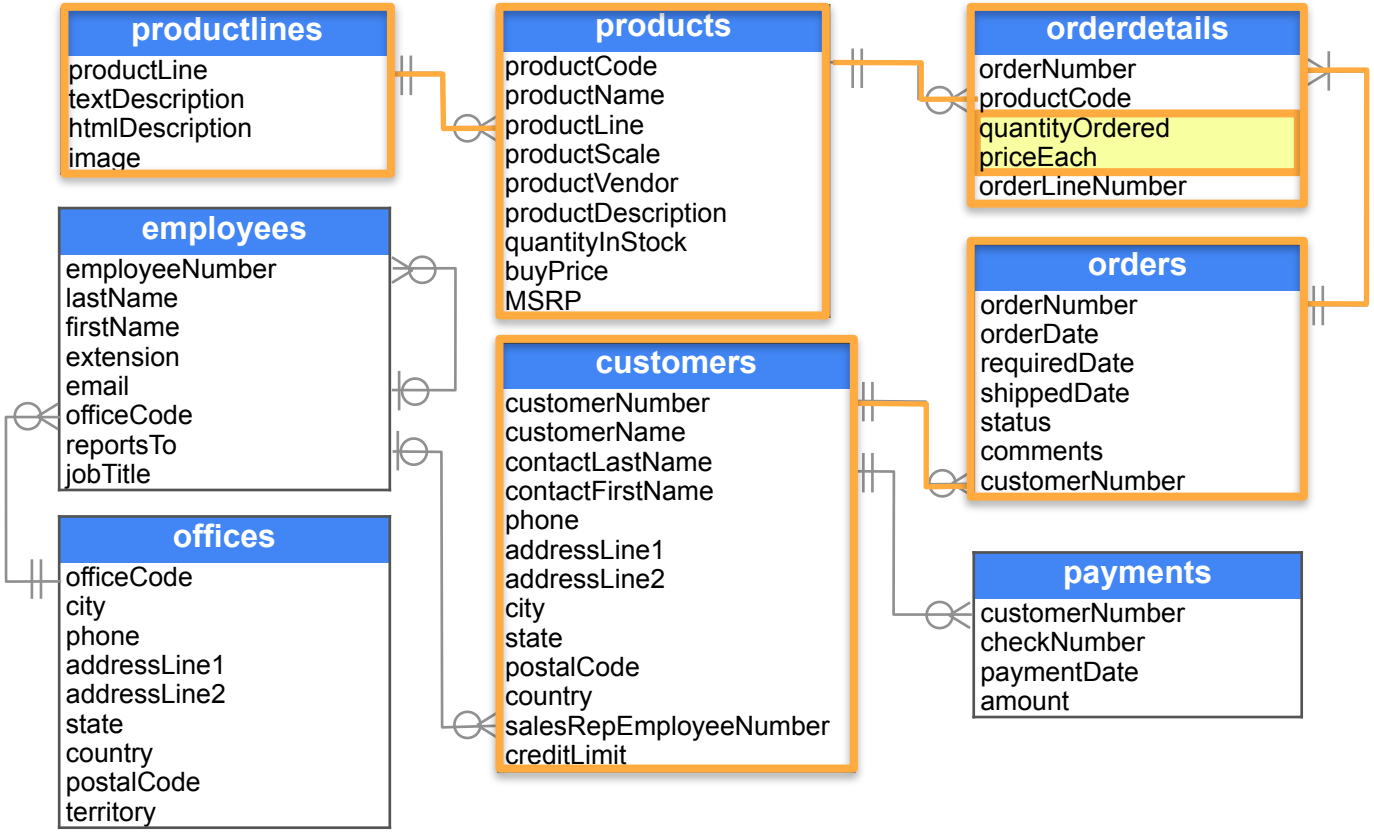


Find **the total sales amount** for each product line within the USA

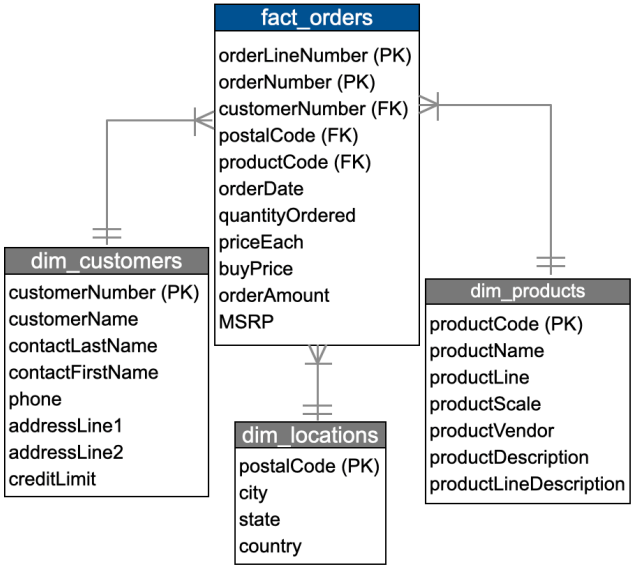
```
SELECT
  dim_products.productLine,
  SUM(fact_orders.orderAmount) AS total_sales
FROM fact_orders
JOIN dim_products ON
  fact_orders.productCode =
  dim_products.productCode
JOIN dim_locations ON
  fact_orders.postalCode =
  dim_locations.postalCode
WHERE dim_locations.country = 'USA'
GROUP BY dim_products.productLine
```

Star Schema VS 3NF

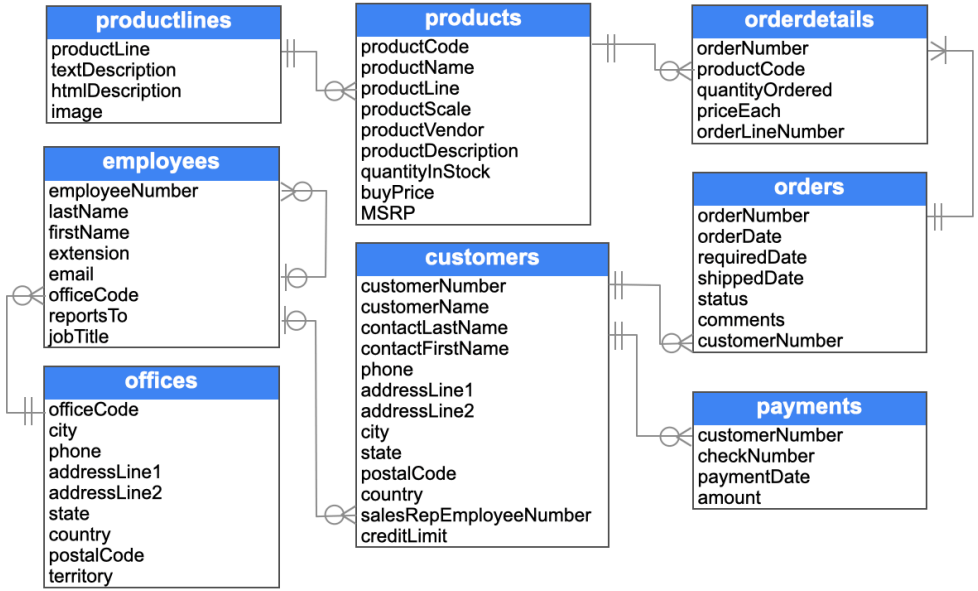
Find **the total sales amount** for each product line within the USA



Star Schema



Normalized Model (3NF)



- Star Schema organizes data so it's easier for business users to understand, navigate and use.
- Star Schema results in simpler queries with fewer joins.

DeepLearning.AI

Data Modeling Techniques

Inmon vs Kimball Data Warehouse Architecture

Inmon Data Modeling Approach

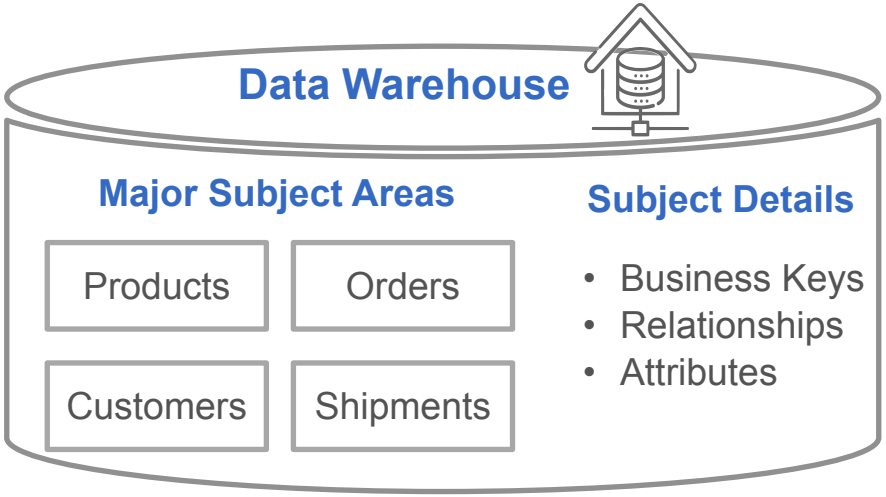
A **subject-oriented**, **integrated**, **nonvolatile**, and **time-variant** collection of data in support of management's decisions.

The data warehouse contains **granular** corporate data. Data in the data warehouse is able to be used for many different purposes, including sitting and waiting for future requirements which are unknown today.

Inmon Data Modeling Approach

A **subject-oriented**, **integrated**, **nonvolatile**, and **time-variant** collection of data in support of management's decisions.

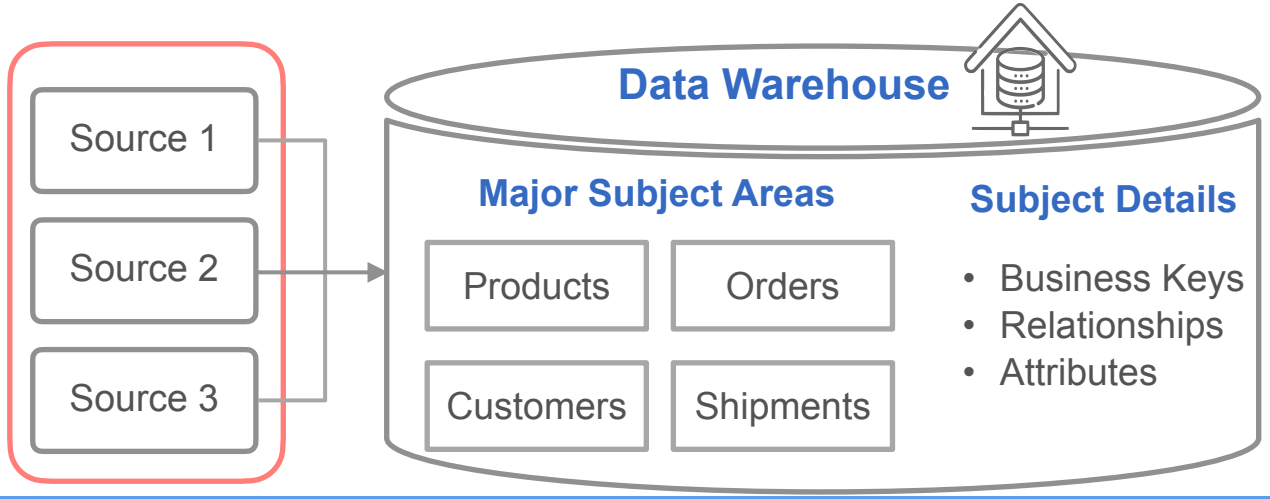
The data warehouse contains **granular** corporate data. Data in the data warehouse is able to be used for many different purposes, including sitting and waiting for future requirements which are unknown today.



Inmon Data Modeling Approach

A **subject-oriented**, **integrated**, **nonvolatile**, and **time-variant** collection of data in support of management's decisions.

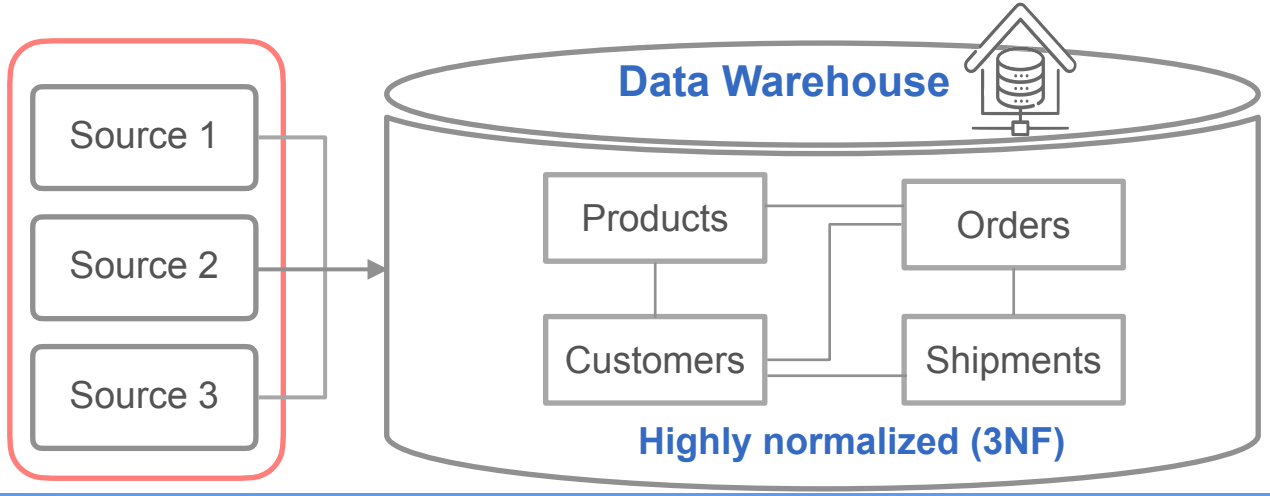
The data warehouse contains **granular** corporate data. Data in the data warehouse is able to be used for many different purposes, including sitting and waiting for future requirements which are unknown today.



Inmon Data Modeling Approach

A **subject-oriented**, **integrated**, **nonvolatile**, and **time-variant** collection of data in support of management's decisions.

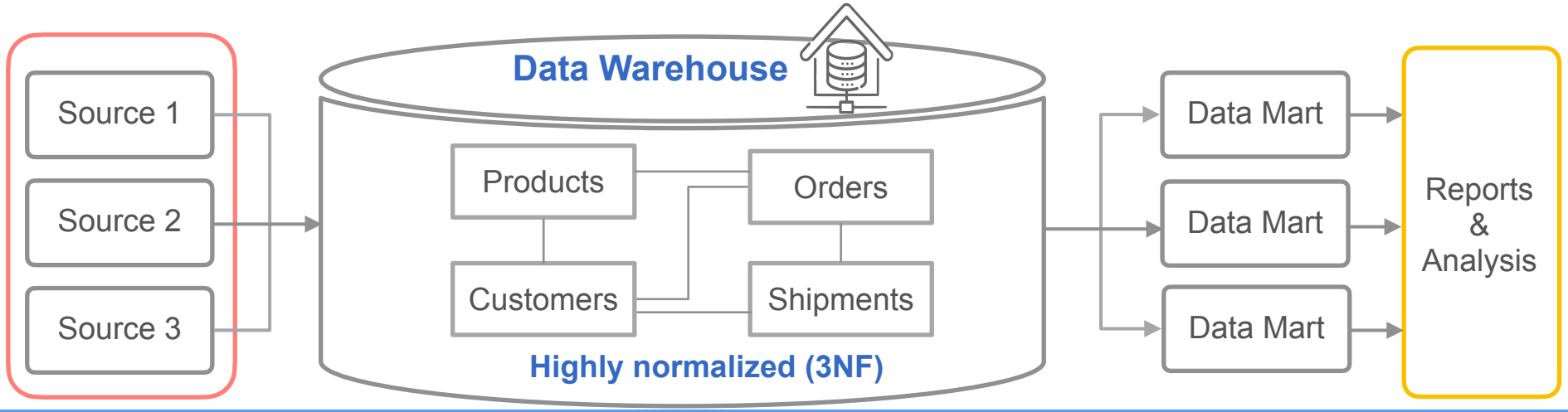
The data warehouse contains **granular** corporate data. Data in the data warehouse is able to be used for many different purposes, including sitting and waiting for future requirements which are unknown today.



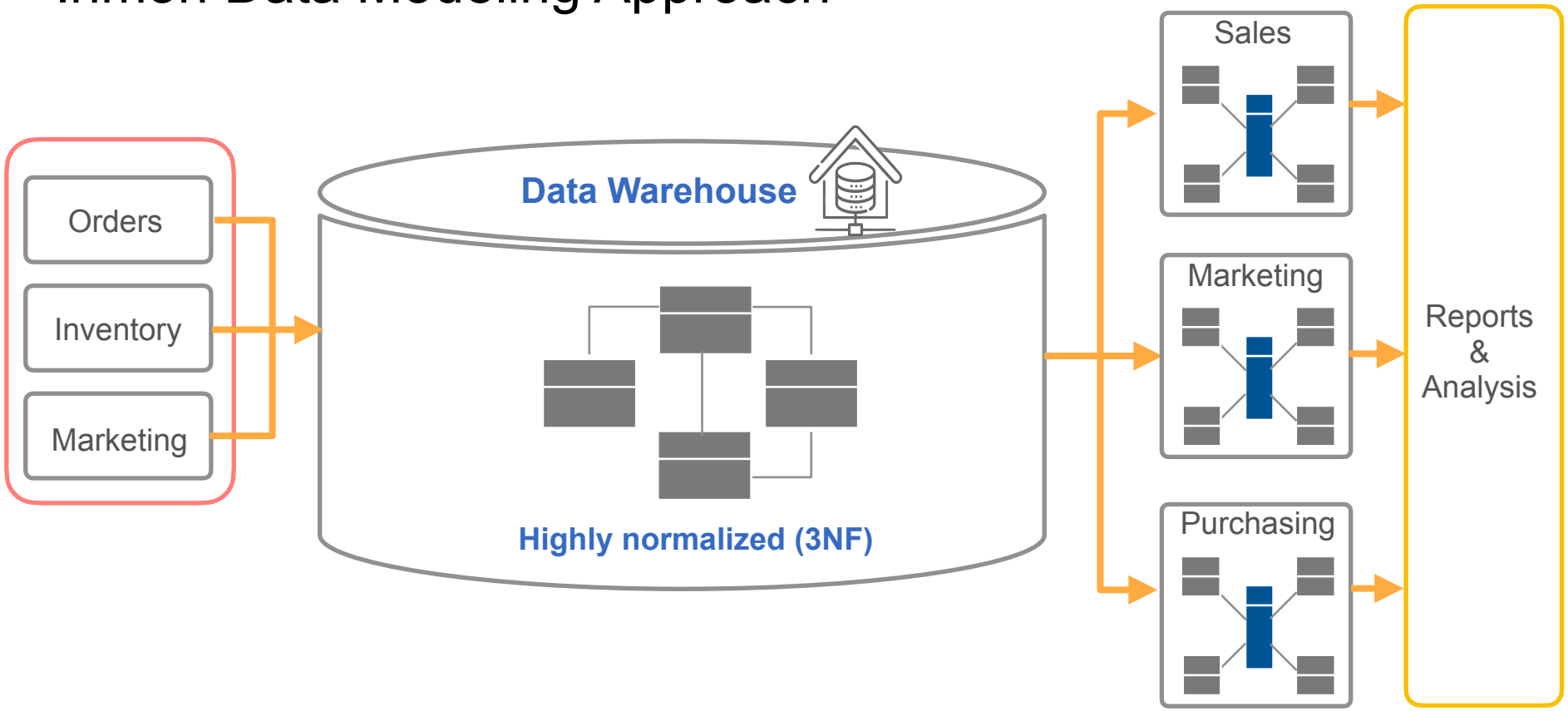
Inmon Data Modeling Approach

A **subject-oriented**, **integrated**, **nonvolatile**, and **time-variant** collection of data in support of management's decisions.

The data warehouse contains **granular** corporate data. Data in the data warehouse is able to be used for many different purposes, including sitting and waiting for future requirements which are unknown today.



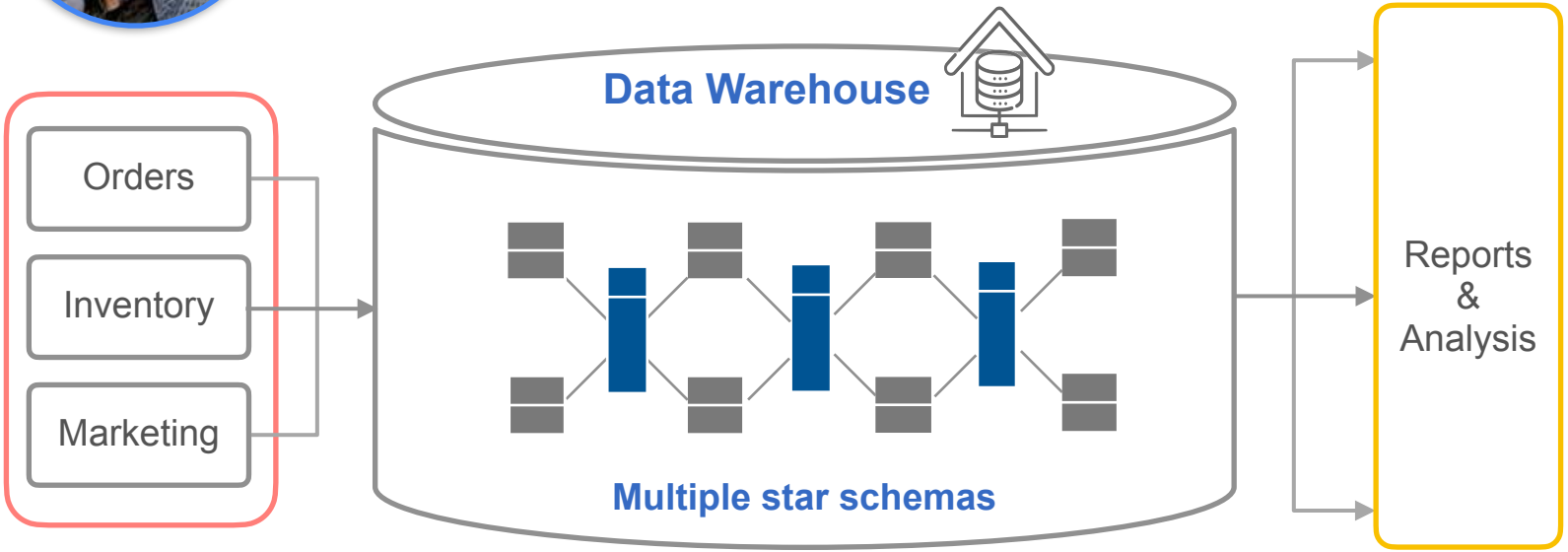
Inmon Data Modeling Approach



Kimball Data Modeling Approach

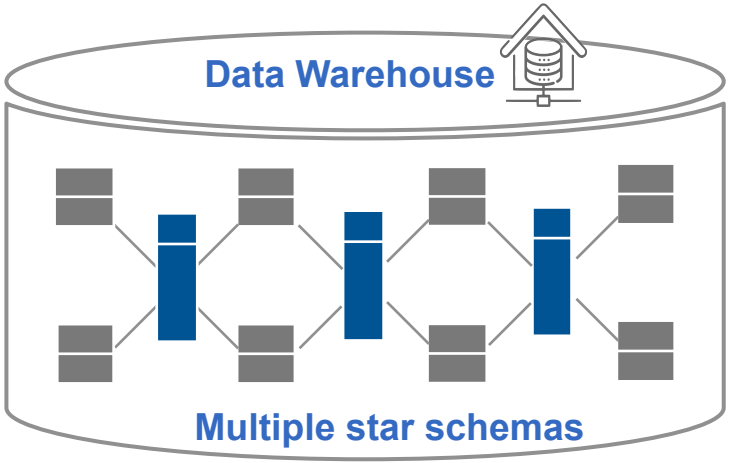
Kimball's approach effectively allows you to serve data that's structured as star schemas (or similar variants) directly from the data warehouse.

- Faster modeling and iteration
- More data redundancy and duplications



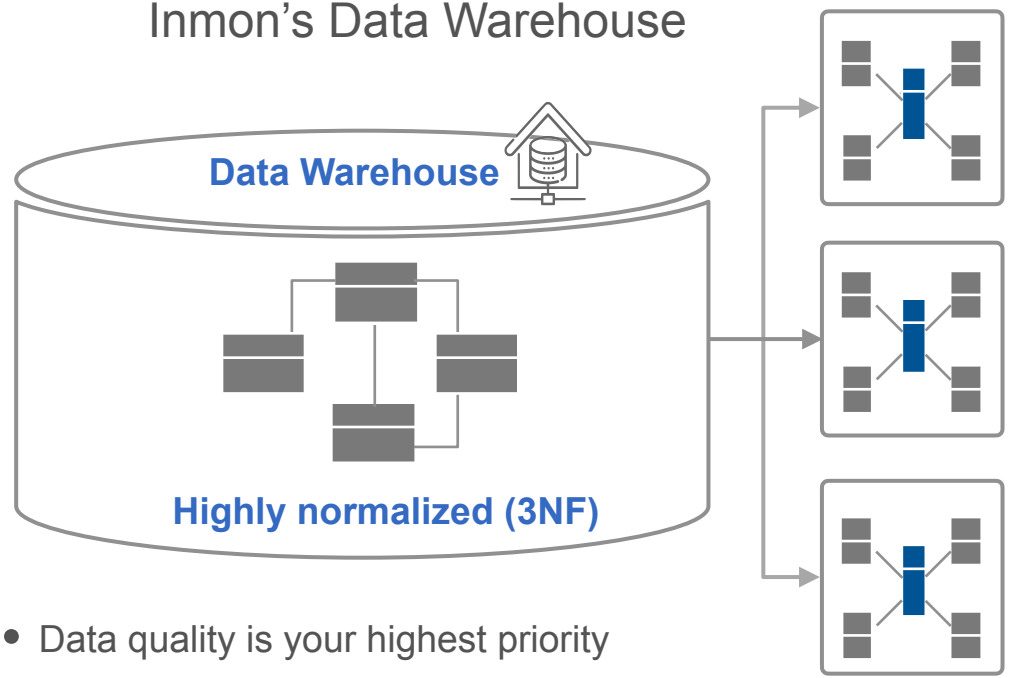
When to Choose Each Approach?

Kimball's Modeling Approach



- Quick insights are your highest priority
- Rapid implementation and iteration

Inmon's Data Warehouse



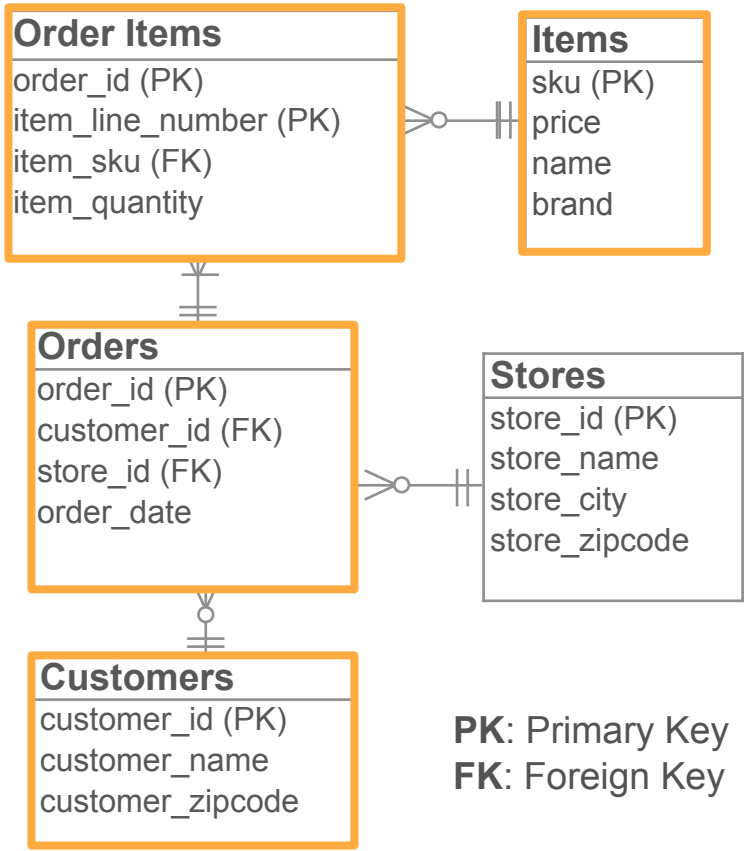
- Data quality is your highest priority
- The analysis requirements are not defined

DeepLearning.AI

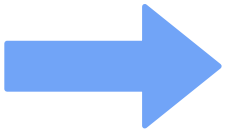
Data Modeling Techniques

Exercise: From Normalization to Star Schema

Normalized Data

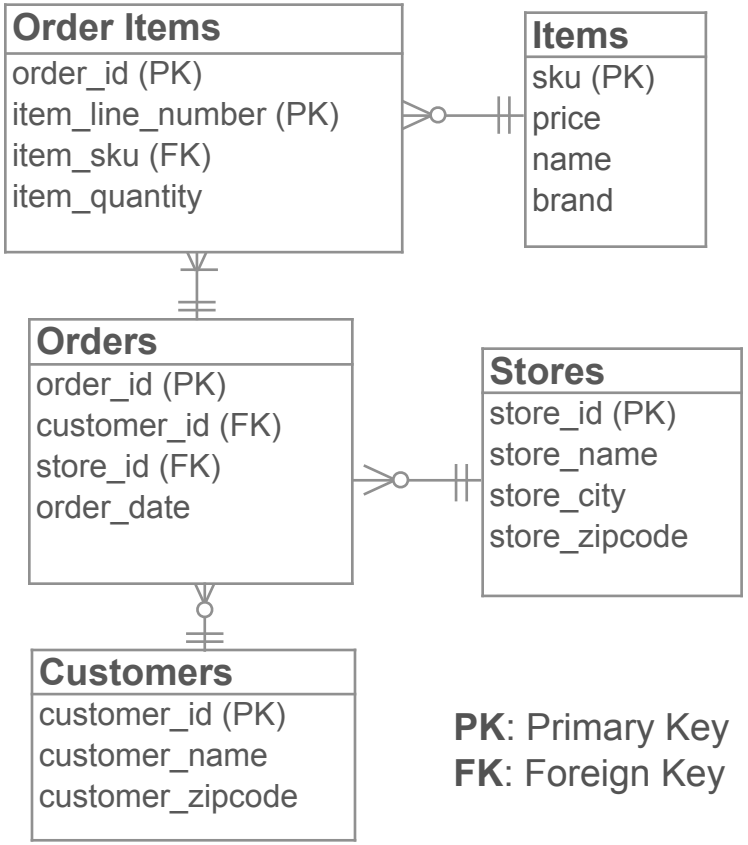


PK: Primary Key
FK: Foreign Key



Star Schema

Normalized Data



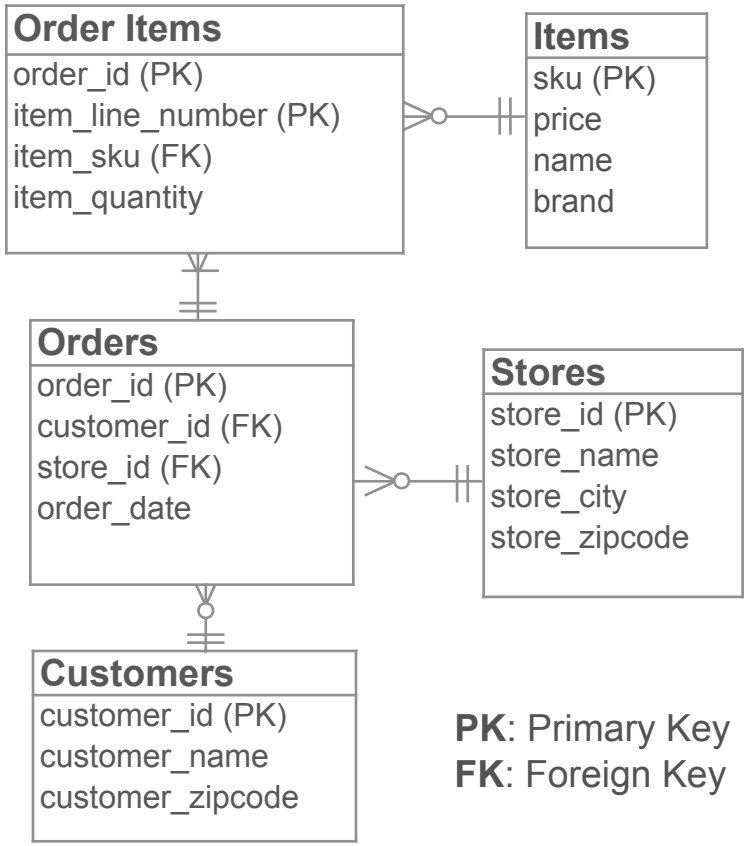
PK: Primary Key
FK: Foreign Key

Star Schema

1. Select the business process → 2. Declare the grain →

Understand the needs of the business

Normalized Data



Star Schema

1. Select the business process
2. Declare the grain
3. Identify the dimensions
4. Identify the facts

Normalized Data

Order Items
order_id (PK)
item_line_number (PK)
item_sku (FK)
item_quantity

Items
sku (PK)
price
name
brand

Orders
order_id (PK)
customer_id (FK)
store_id (FK)
order_date

Stores
store_id (PK)
store_name
store_city
store_zipcode

Customers
customer_id (PK)
customer_name
customer_zipcode

PK: Primary Key
FK: Foreign Key

Star Schema

1. Select the business process
2. Declare the grain

Analyze the sales data:

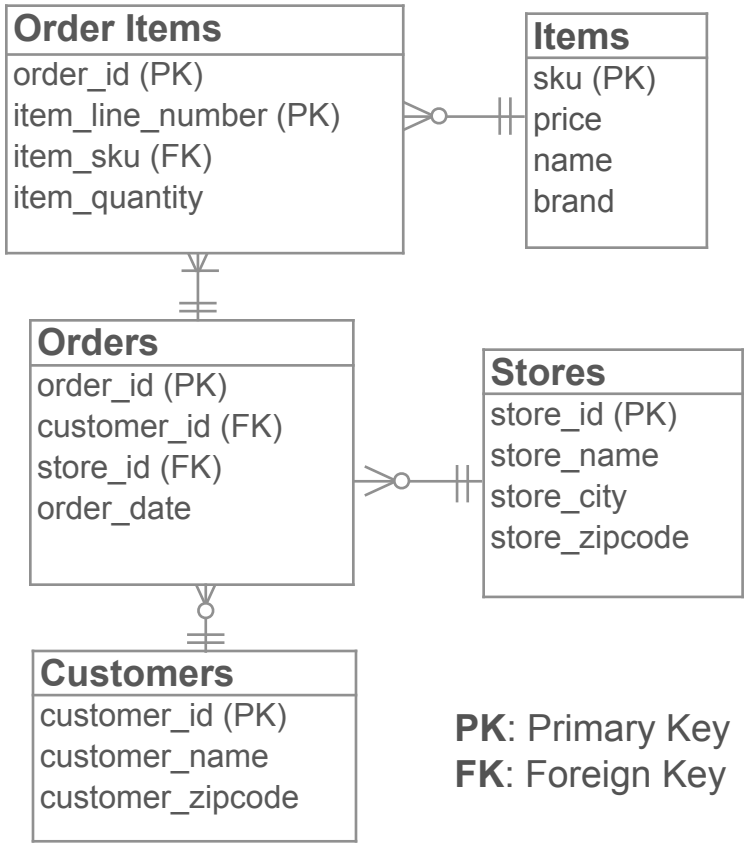
- which products are selling in which stores on a given day
- differences in the sales between the stores
- which product brands are most popular

Business process: company's sales transactions

Atomic Grain:

- Total sales transactions on a particular day
- Single sales transaction
- Individual product item in a sales transaction

Normalized Data



PK: Primary Key
FK: Foreign Key

Star Schema

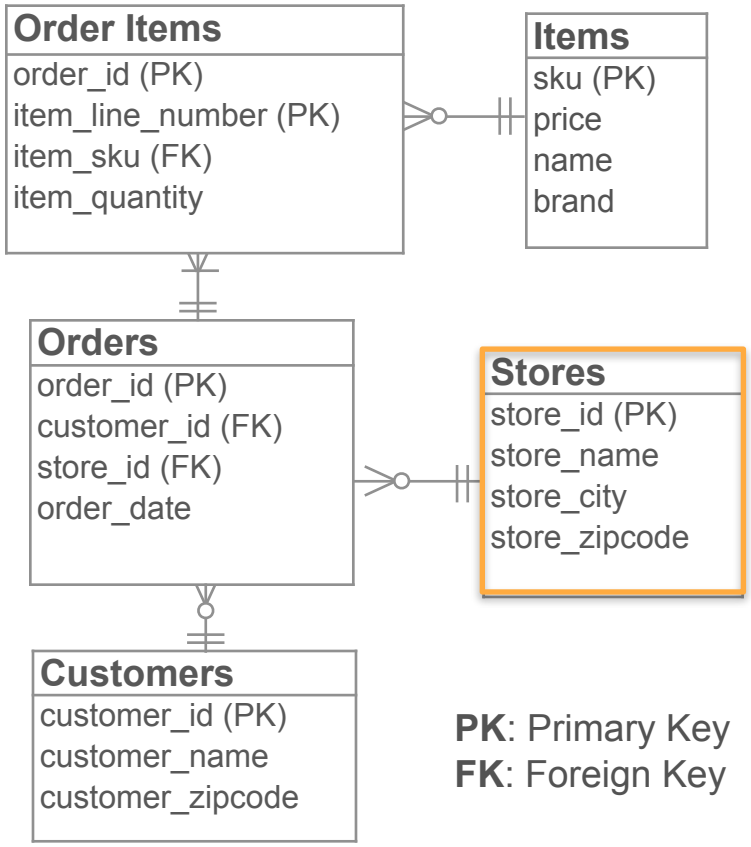
Bus. process: sales transactions → Grain: individual product in a transaction → 3. Identify the dimensions →

Analyze the sales data:

- which products are selling in which stores on a given day
- differences in the sales between the stores
- which product brands are most popular

dim_stores	dim_items	dim_date

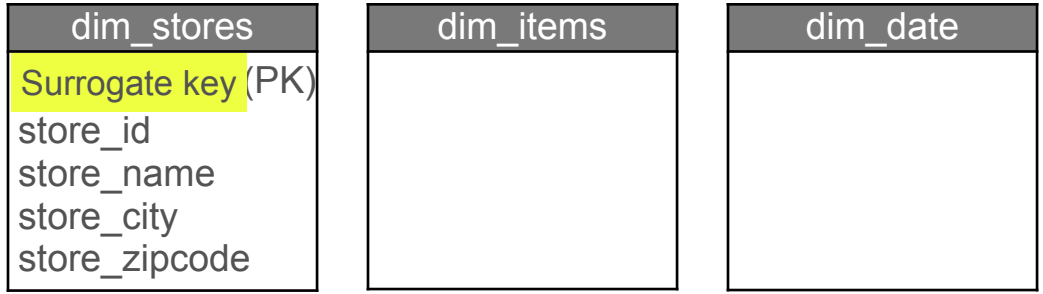
Normalized Data



PK: Primary Key
FK: Foreign Key

Star Schema

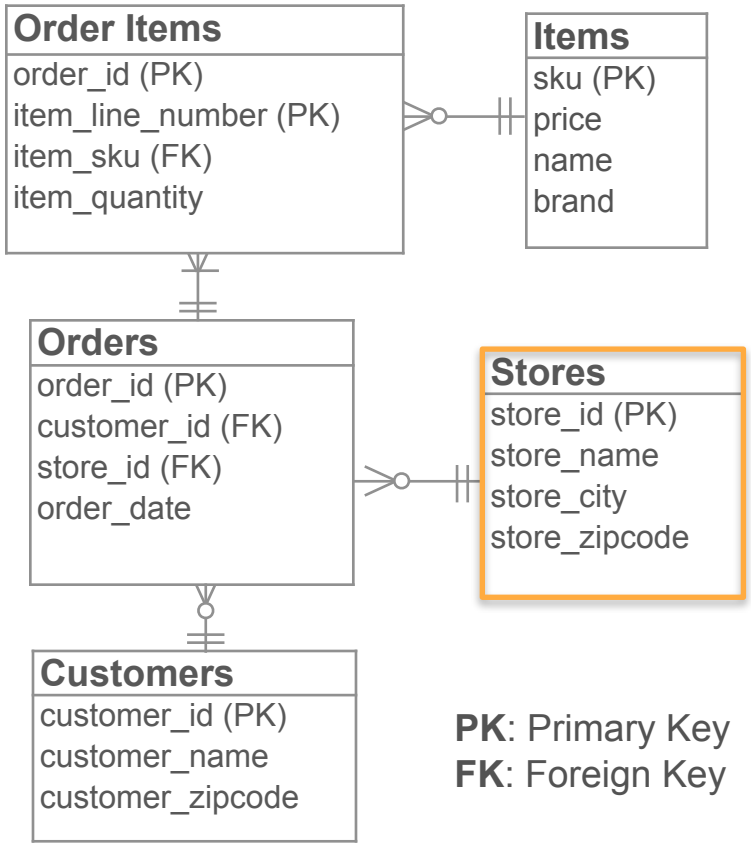
Bus. process: sales transactions → Grain: individual product in a transaction → 3. Identify the dimensions →




```
SELECT store_id,
       store_name,
       store_city,
       store_zipcode

FROM stores;
```

Normalized Data



PK: Primary Key
FK: Foreign Key

Star Schema

Bus. process: sales transactions → Grain: individual product in a transaction → 3. Identify the dimensions →

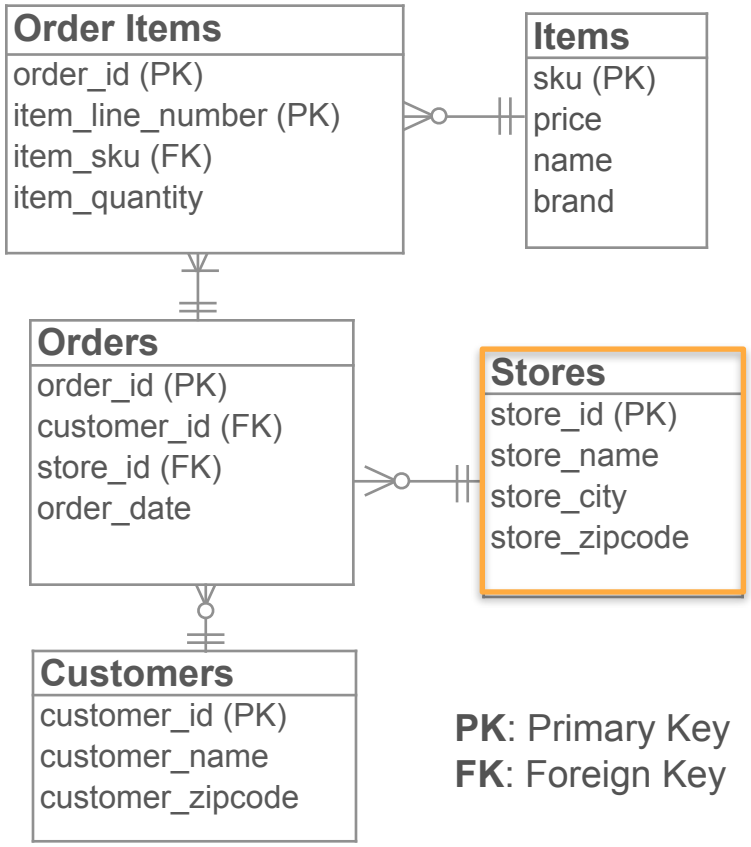
dim_stores
Surrogate key (PK)
store_id
store_name
store_city
store_zipcode

- Create a sequence of integers and assign one integer to each store
- Use a hash function to generate a unique surrogate key
  - Supported by many DBMSs like PostgreSQL and MySQL
  - Example: MD5

```
SELECT store_id,
       store_name,
       store_city,
       store_zipcode

FROM stores;
```

Normalized Data



PK: Primary Key
FK: Foreign Key

Star Schema

Bus. process: sales transactions → Grain: individual product in a transaction → 3. Identify the dimensions →

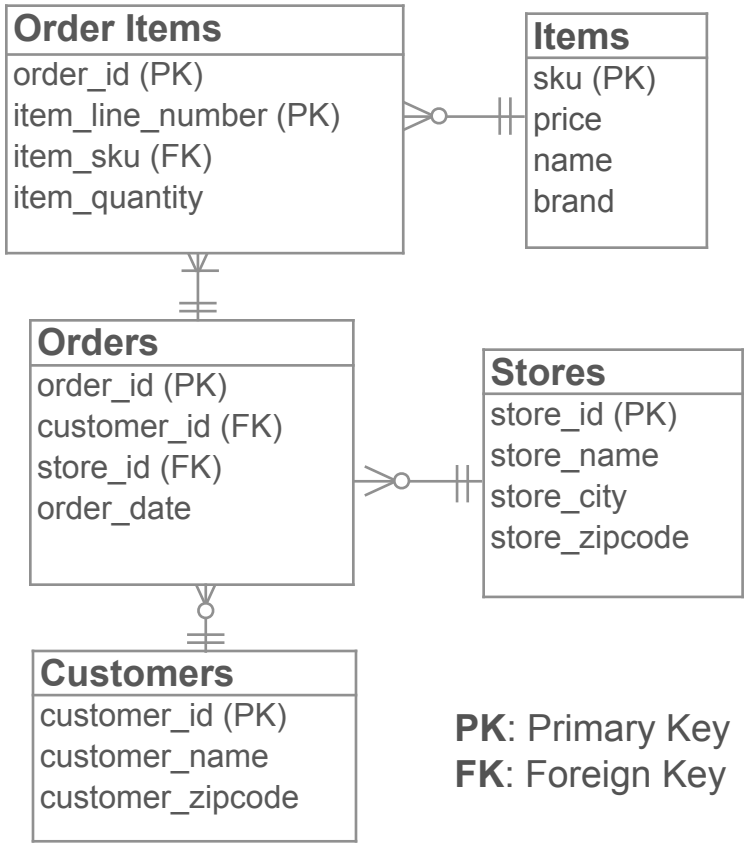
dim_stores	
store_key	(PK)
store_id	
store_name	
store_city	
store_zipcode	

- Create a sequence of integers and assign one integer to each store
- Use a hash function to generate a unique surrogate key
  - Supported by many DBMSs like PostgreSQL and MySQL
  - Example: MD5

```
SELECT MD5(store_id) as store_key,
       store_id,
       store_name,
       store_city,
       store_zipcode

FROM stores;
```

Normalized Data



PK: Primary Key
FK: Foreign Key

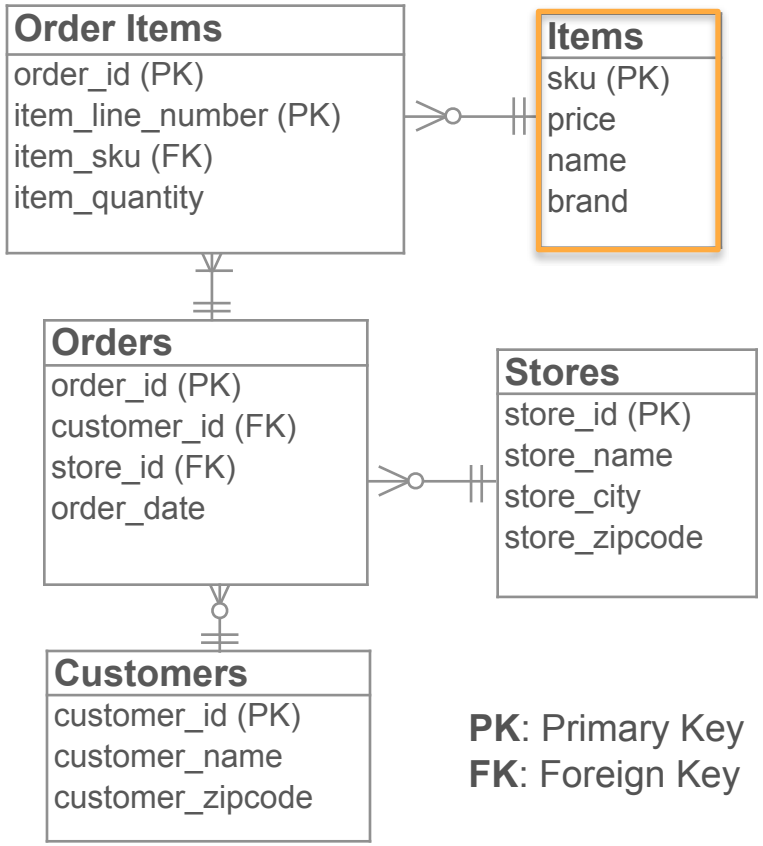
Star Schema

Bus. process: sales transactions → Grain: individual product in a transaction → 3. Identify the dimensions


```
SELECT MD5(store_id) as store_key,
       store_id,
       store_name,
       store_city,
       store_zipcode

FROM stores;
```

Normalized Data



PK: Primary Key
FK: Foreign Key

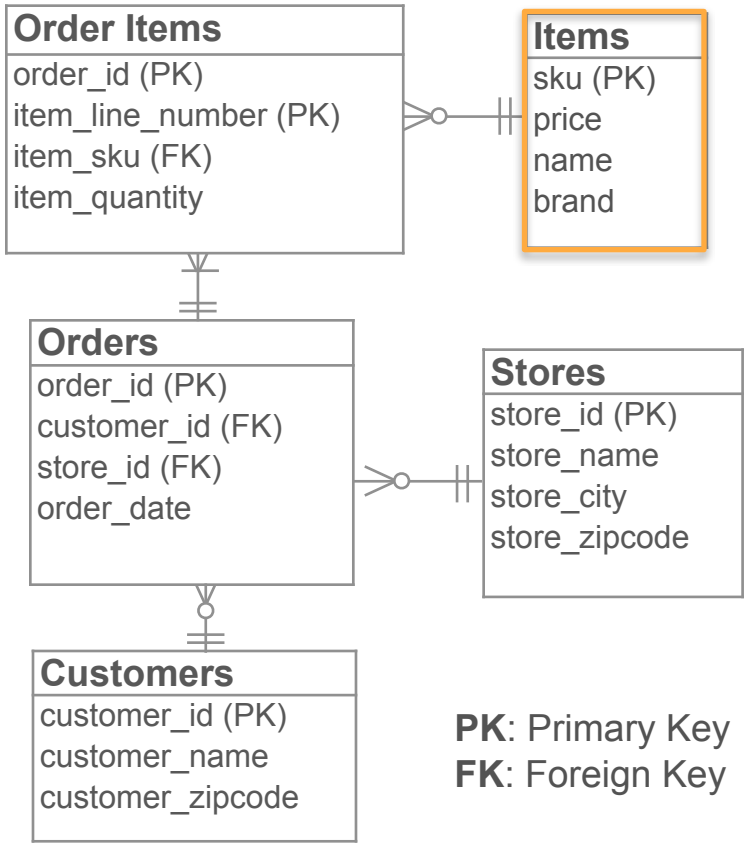
Star Schema

Bus. process: sales transactions → Grain: individual product in a transaction → 3. Identify the dimensions →

dim_stores	dim_items	dim_date
store_key (PK)	sku	
store_id	name	
store_name	brand	
store_city		
store_zipcode		

```
SELECT sku,
       name,
       brand
FROM items;
```

Normalized Data



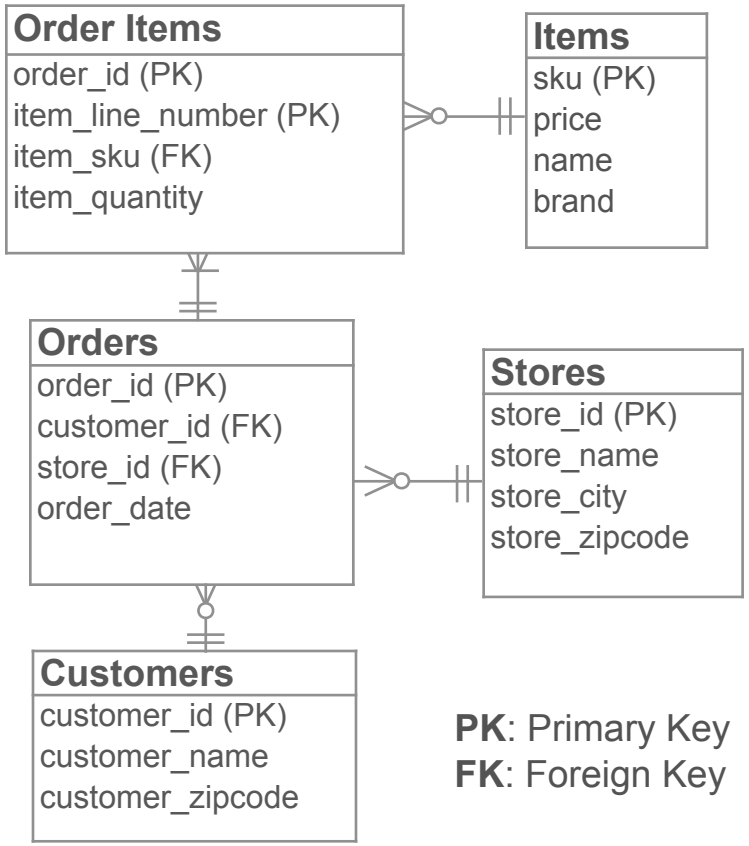
PK: Primary Key
FK: Foreign Key

Star Schema

Bus. process: sales transactions → Grain: individual product in a transaction → 3. Identify the dimensions →


```
SELECT MD5(sku) as item_key,
       sku,
       name,
       brand
FROM items;
```

Normalized Data



PK: Primary Key
FK: Foreign Key

Star Schema

Bus. process: sales transactions → Grain: individual product in a transaction → 3. Identify the dimensions →

dim_stores
store_key (PK)
store_id
store_name
store_city
store_zipcode

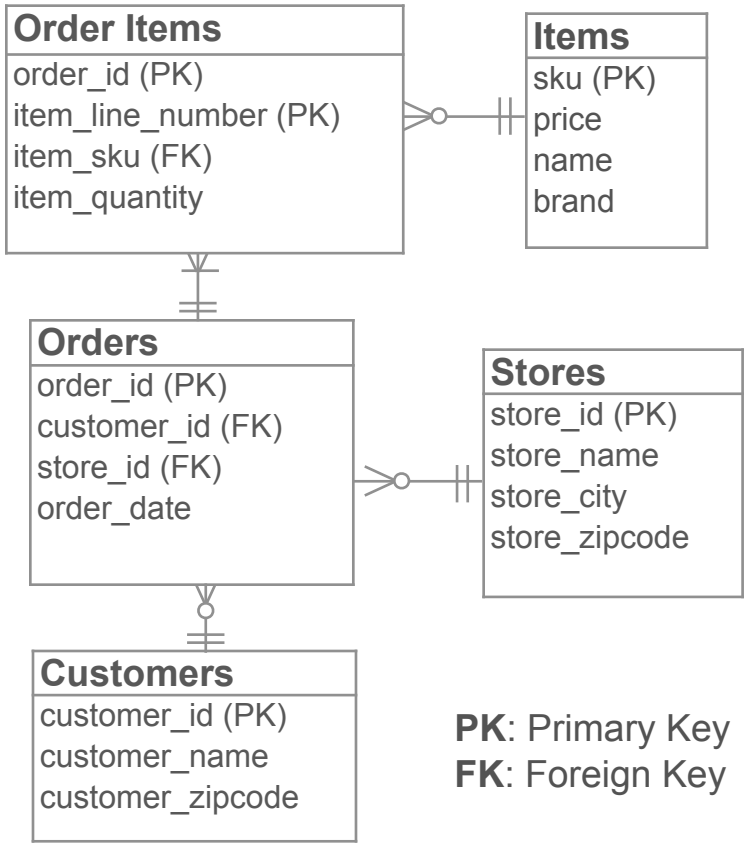
dim_items
item_key (PK)
sku
name
brand

dim_date

Date	Year	Quarter	Month	Day-of-week
2022-03-01	2022	1	3	Tuesday
2022-03-02	2022	1	3	Wednesday

- What are the total sales in the first quarter of 2022?
- What products are more popular on the weekends?

Normalized Data



PK: Primary Key
FK: Foreign Key

Star Schema

Bus. process: sales transactions → Grain: individual product in a transaction → 3. Identify the dimensions

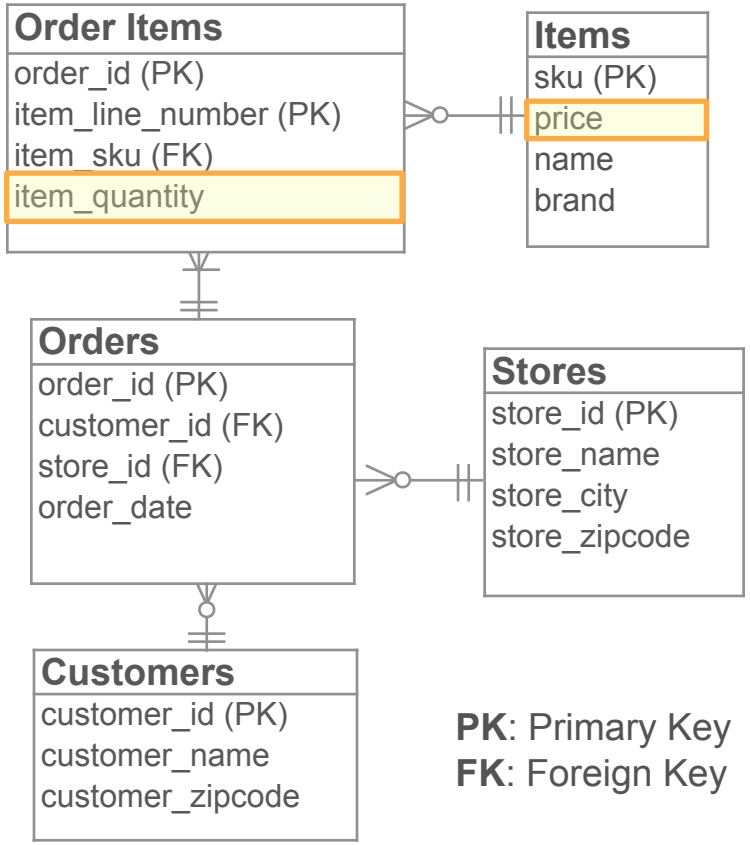
dim_stores
store_key (PK)
store_id
store_name
store_city
store_zipcode

dim_items
item_key (PK)
sku
name
brand

dim_date
date_key(PK)
day_of_week
month
quarter
Year

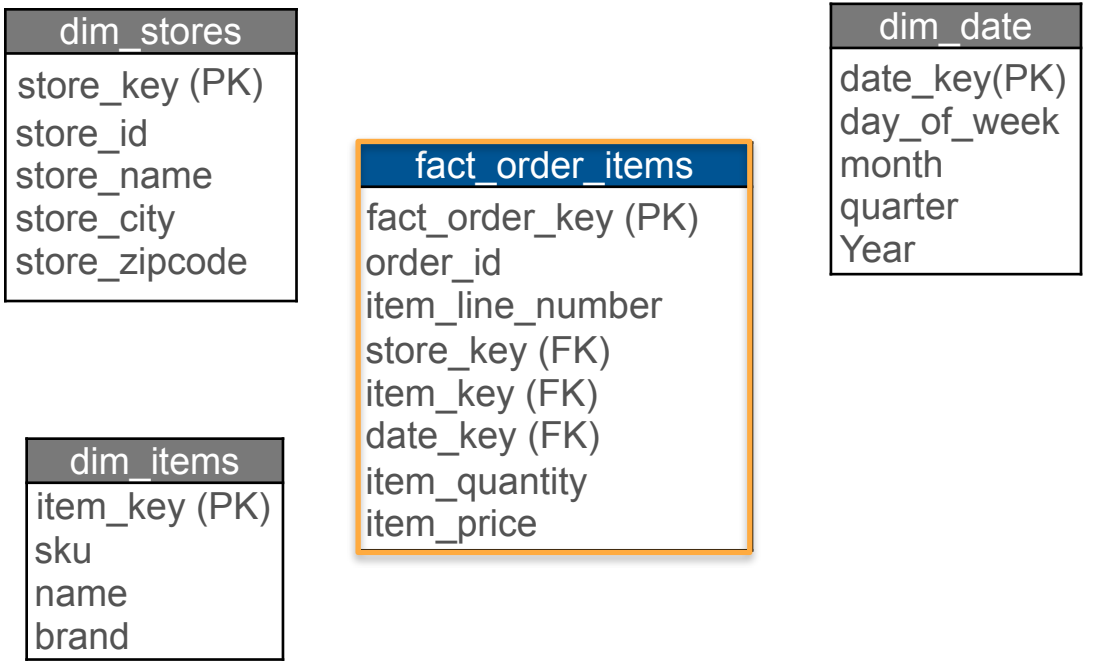
```
SELECT date_key,
  EXTRACT(DAY FROM date_key) AS day_of_week,
  EXTRACT(MONTH FROM date_key) AS month,
  EXTRACT(Quarter FROM date_key) AS quarter,
  EXTRACT(year FROM date_key) AS YEAR
FROM generate_series ('2020-01-01'::date,
                      '2025-01-01'::date,
                      '1 day'::interval) As date_key
```

Normalized Data

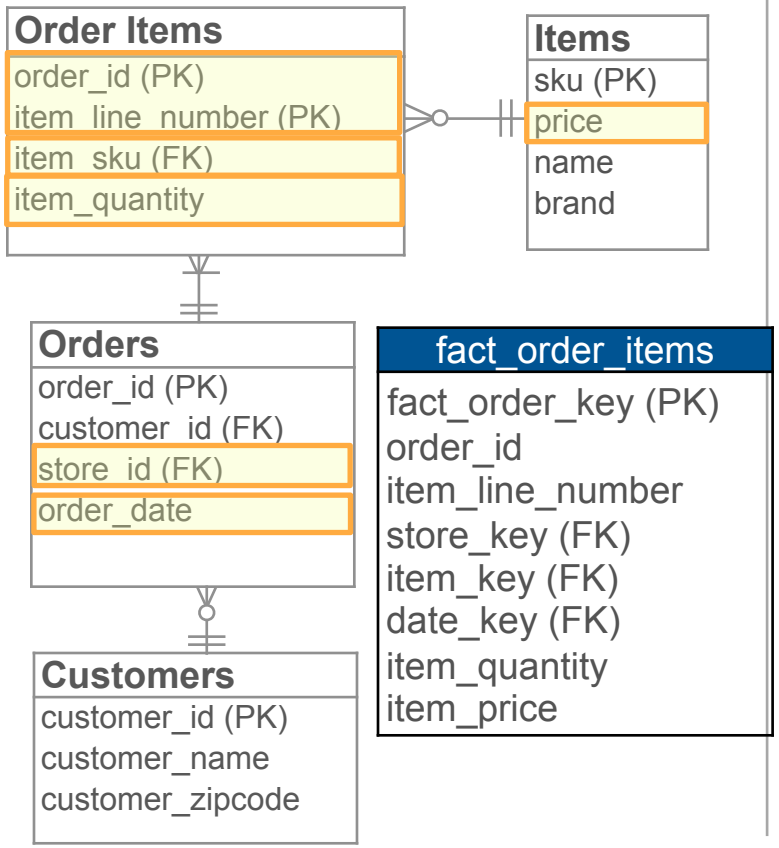


Star Schema

Bus. process: sales transactions → Grain: individual product in a transaction → 3. Identify the dimensions → 4. Identify the facts



Normalized Data



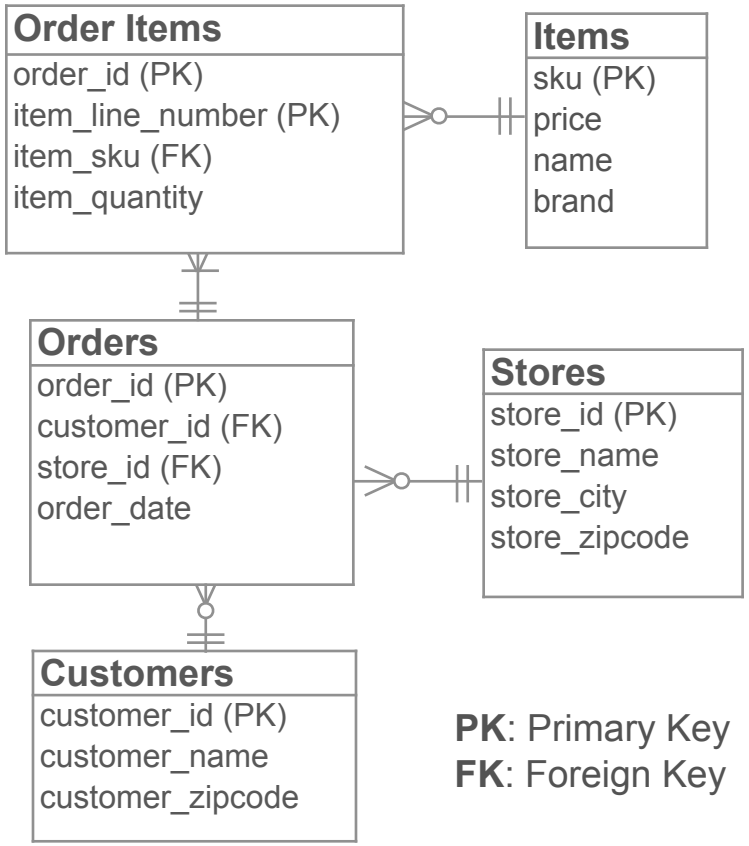
Star Schema

Bus. process: sales transactions → Grain: individual product in a transaction → Identify the dimensions → Identify the facts

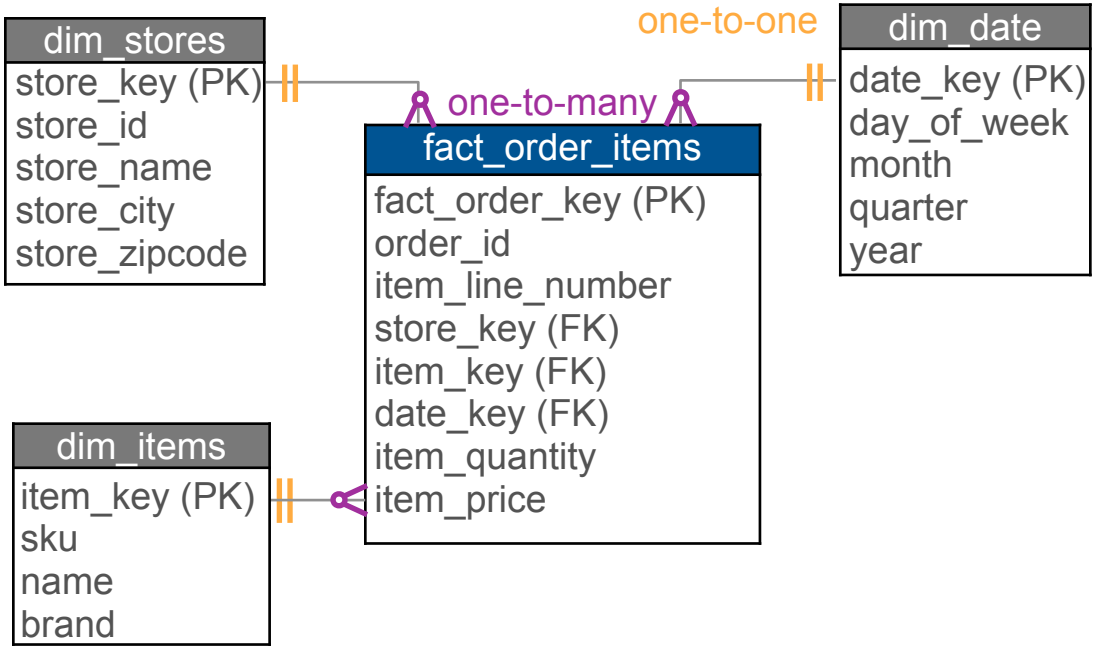
```
Select
   MD5(CONCAT(OrderItems.order_id,
              OrderItems.item_line_number ))
   AS fact_order_key,
   OrderItems.order_id,
   OrderItems.item_line_number,
   MD5(Orders.store_id) AS store_key,
   MD5(OrderItems.item_sku) AS item_key,
   Orders.order_date AS date_key,
   OrderItems.item_quantity,
   Items.price AS item_price

FROM OrderItems
Join Orders ON Orders.order_id = OrderItems.order_id
Join Items ON Items.sku = OrderItems.item_sku
```

Normalized Data

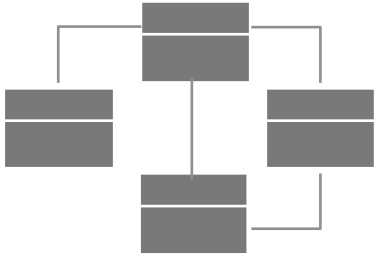


Star Schema



Week 1 Lab 2

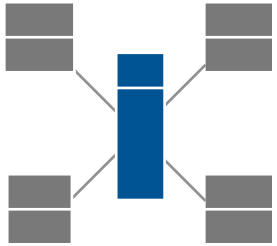
Normalized Data



Model the data

dbt

Star Schema



dbt

- Connects to your data warehouse
- Transforms and validates your data within the data warehouse
- Generates the SQL code behind the scenes to transform your data
- Can't join together data from different sources or move transformed data to another target system

AWS Glue

- Can connect to different sources, apply transformations, and store processed data somewhere else

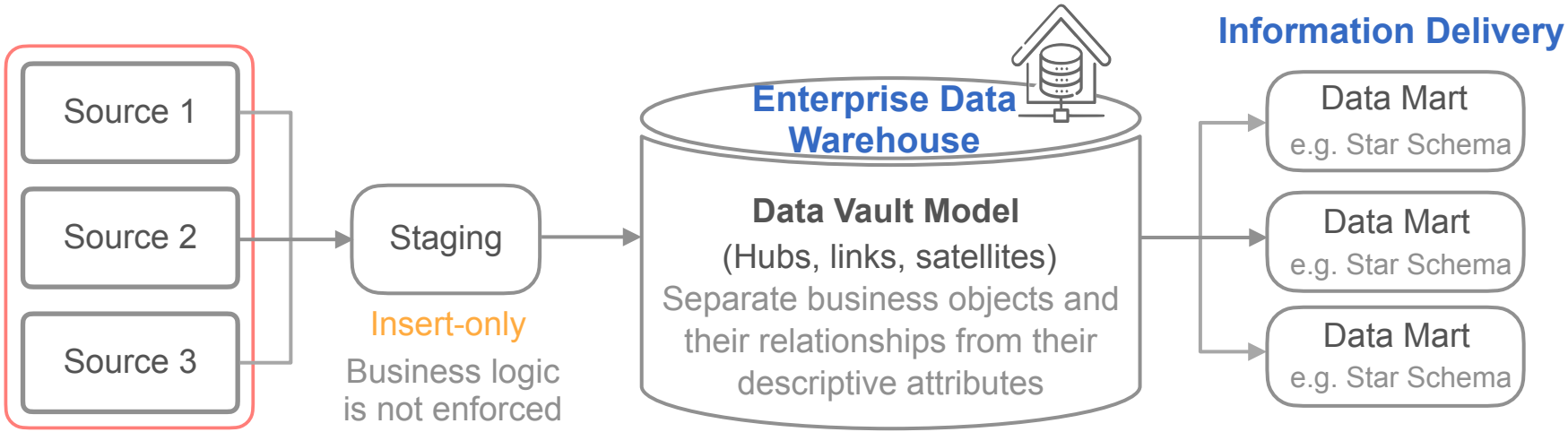
DeepLearning.AI

Data Modeling Techniques

Data Vault

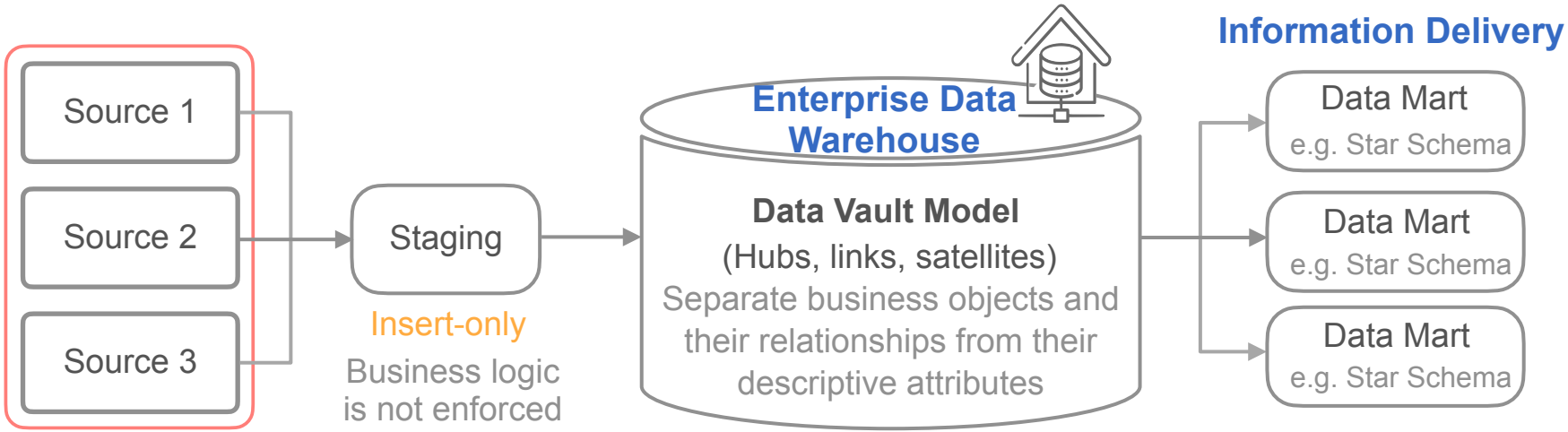
Data Vault

Dan Linstedt introduced Data Vault as a different approach to modeling the data in the data warehouse.



Data Vault

- No notion of good, bad, or conformed data in a data vault
- Only change the structure in which data is stored:
  - Allows you to trace the data back to its source
  - Helps you avoid restructuring the data when business requirements change



Data Vault Model

Three main types of tables:

Hub

Stores a unique list of business keys

Customers, products, employees, vendors

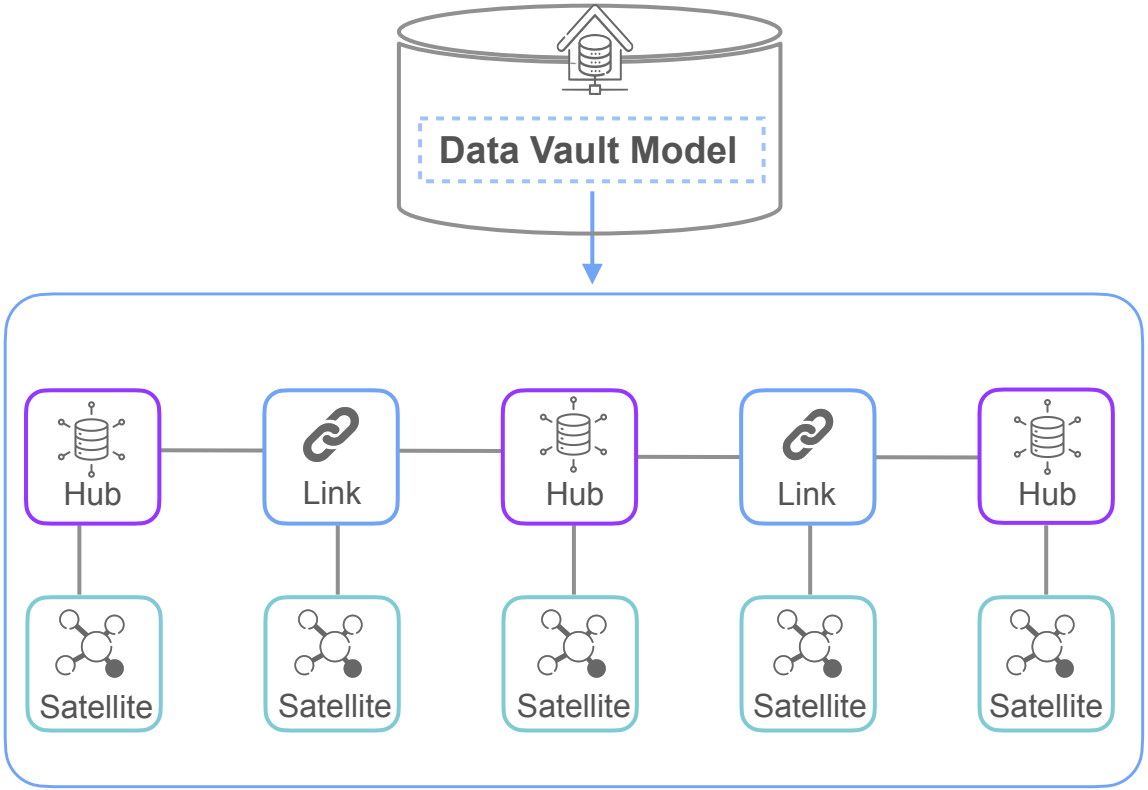
Link

Connects two or more hubs

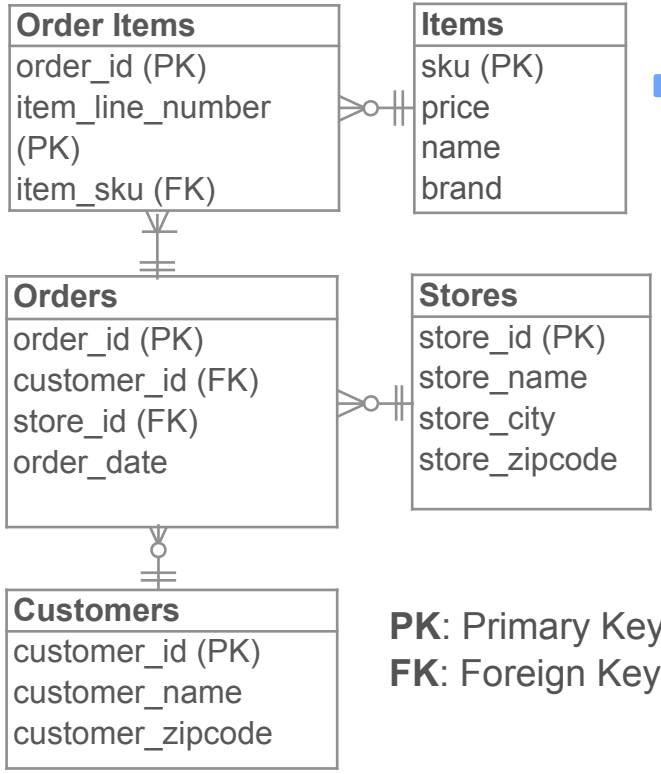
Relationship, transaction, event

Satellite

Contains attributes that provide context for hubs and links



Data Vault - Step 1: Model the Hubs

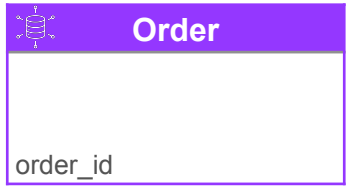


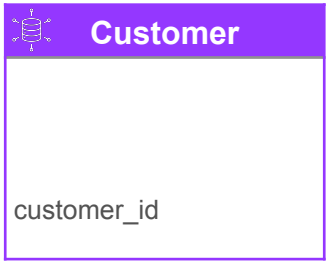
Data Vault

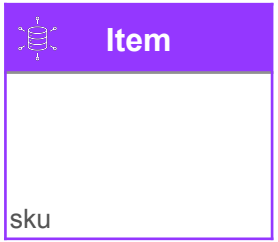
- What is the identifiable business element?
- How do users commonly look for data?
- A business key:
  - column(s) used by the business to identify and locate the data
  - not be a key generated in or tied to a particular source system

Data Vault - Step 1: Model the Hubs

Data Vault



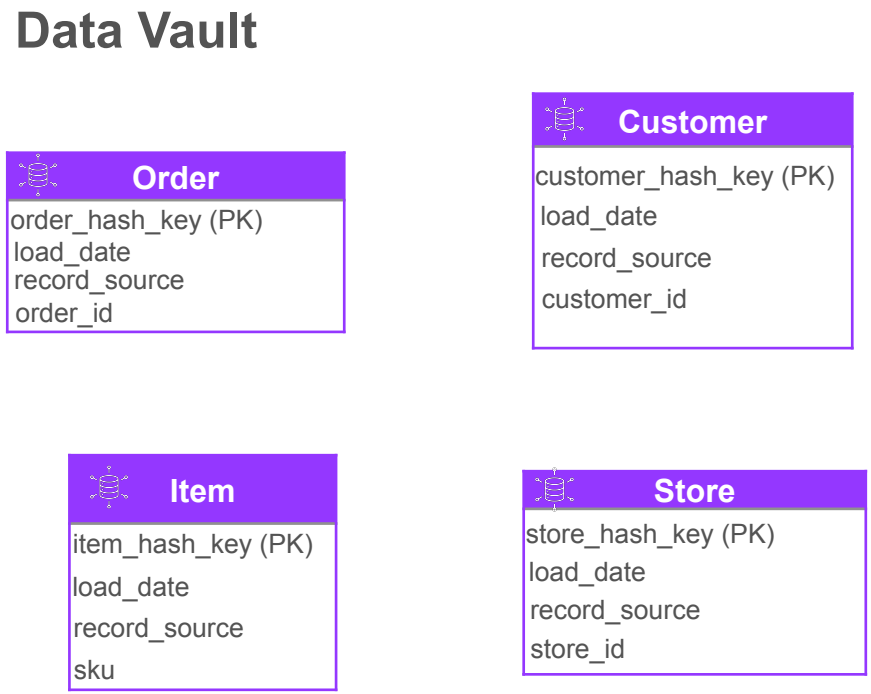




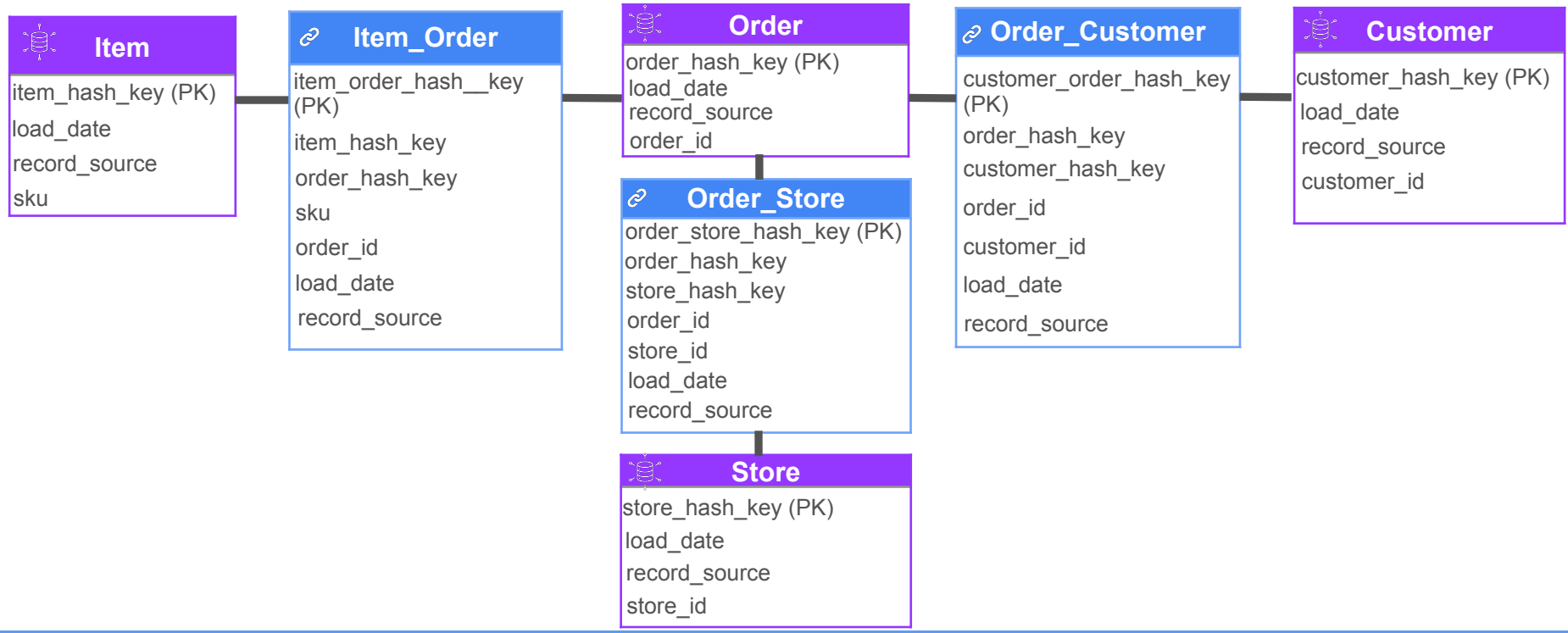
Data Vault - Step 1: Model the Hubs

A hub should contain:

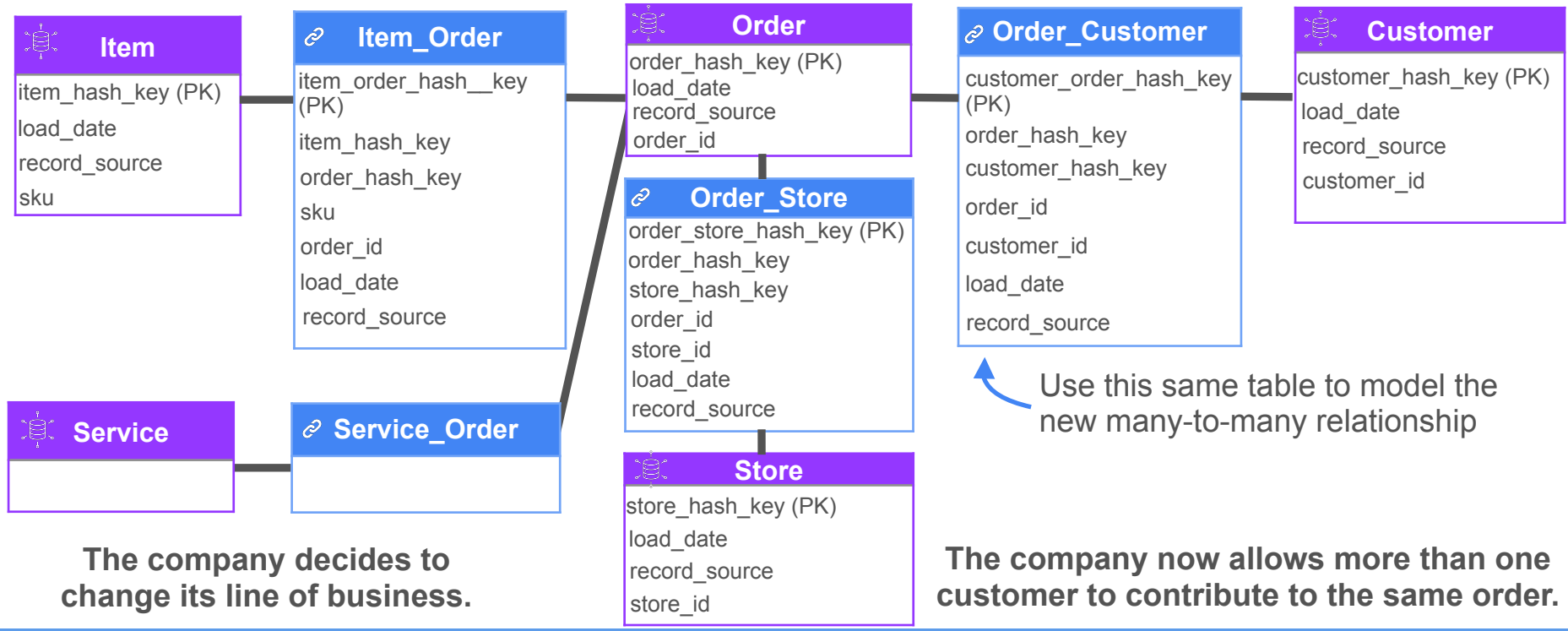
- **The business key**
- **The hash key**:
  - Calculated as a hash of the business key
  - Used as the Hub primary key
- **The load date**: date on which the business key was first loaded
- **The record source**: the source of the business key



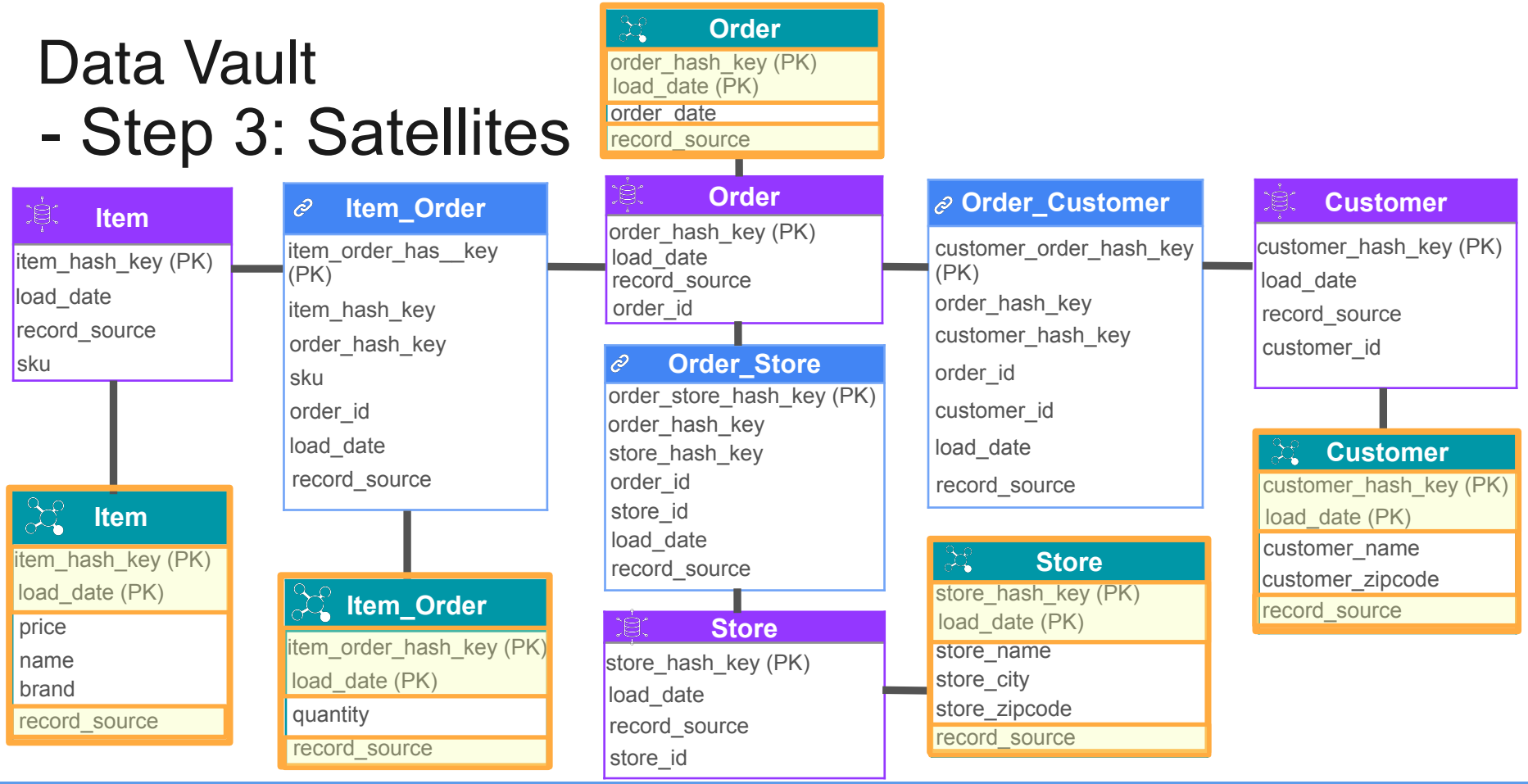
Data Vault - Step 2: Model the Links



Data Vault - Step 2: Model the Links



Data Vault
- Step 3: Satellites



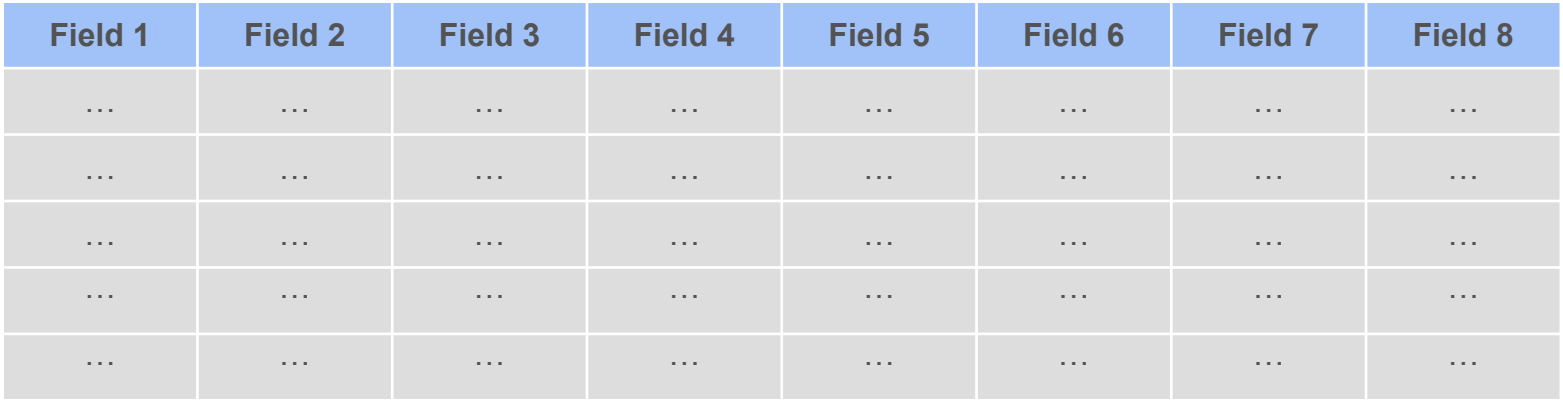
DeepLearning.AI

Data Modeling Techniques

One Big Table

One Big Table (OBT)

Many columns
(Can be thousands of columns)



Field 1	Field 2	Field 3	Field 4	Field 5	Field 6	Field 7	Field 8
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…

Highly denormalized and flexible

Single value or nested data

Wide Table Example

- Can have hundreds or more columns
- Combines various data types

OrderID	Orderitems	CustomerID	CustomerName	address	OrderDate

100	[{
	“sku”:1,
	“price”:50,
	“quantity”: 1,
	“name”: “Thingamajig”
	}, {
	“sku”:2,
	“price”:25,
	“quantity”: 3,
	“name”: “Whatchamacallit”
	}]
		5	Joe Reis	1st. St	1/08/2024

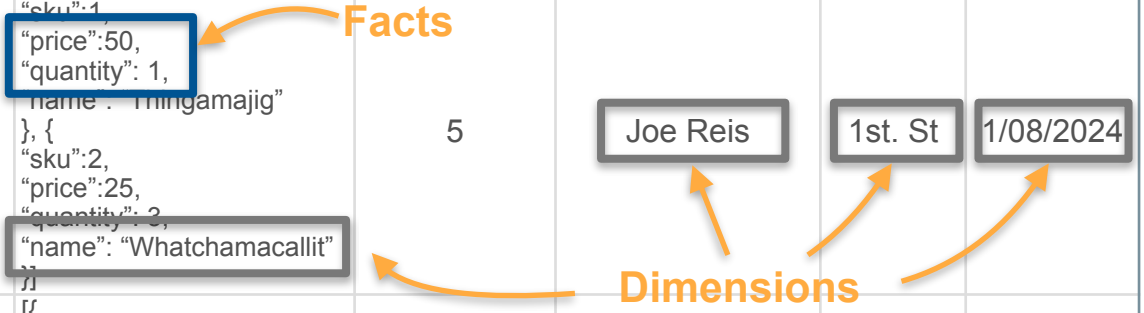
101	[{
	“sku”:3,
	“price”:75,
	“quantity”: 1,
	“name”: “Whoozeewhatzit”
	}, {
	“sku”:2,
	“price”:25,
	“quantity”: 3,
	“name”: “Whatchamacallit”
	}]
		7	Matt Housely	2nd Ave.	1/08/2024

102	[{ “sku”:1, “price”:50, “quantity”: 1, “name”: “Thingamajig” }]	9	Colleen Fotsch	2nd Ave.	1/08/2024

Wide Table Example

- Can have hundreds or more columns
- Combines various data types
- No need for complex joins
- Supports fast analytical queries

OrderID	Orderitems	CustomerID	CustomerName	address	OrderDate
100	[{ “sku”:1, “price”:50, “quantity”: 1, “name”: “Thingamajig” }, { “sku”:2, “price”:25, “quantity”: 3, “name”: “Whatchamacallit” }]	5	Joe Reis	1st. St	1/08/2024
101	[{ “sku”:3, “price”:75, “quantity”: 1, “name”: “Whoozeewhatzit” }, { “sku”:2, “price”:25, “quantity”: 3, “name”: “Whatchamacallit” }]	7	Matt Housely	2nd Ave.	1/08/2024
102	[{ “sku”:1, “price”:50, “quantity”: 1, “name”: “Thingamajig” }]	9	Colleen Fotsch	2nd Ave.	1/08/2024



Why are OBTs becoming popular?

- Low cost of cloud storage 
- Nested data allows for flexible schemas
- Columnar storage helps optimize the storage and processing of OBTs:
  - Wide tables are sparse
  - Columnar database reads only columns selected in a query, and reading nulls is essentially free

Order ID	Price	Product SKU	Quantity	Customer ID
1	40	45865	10	67t
2	23	90234	14	56t
3	45	12558	12	87q
4	50	45682	13	98q
...	...	...	...	...

Cons

- You might lose the business logic in your analytics
- You need complex data structures to store nested data:
  - Can have poorer update and aggregation performance

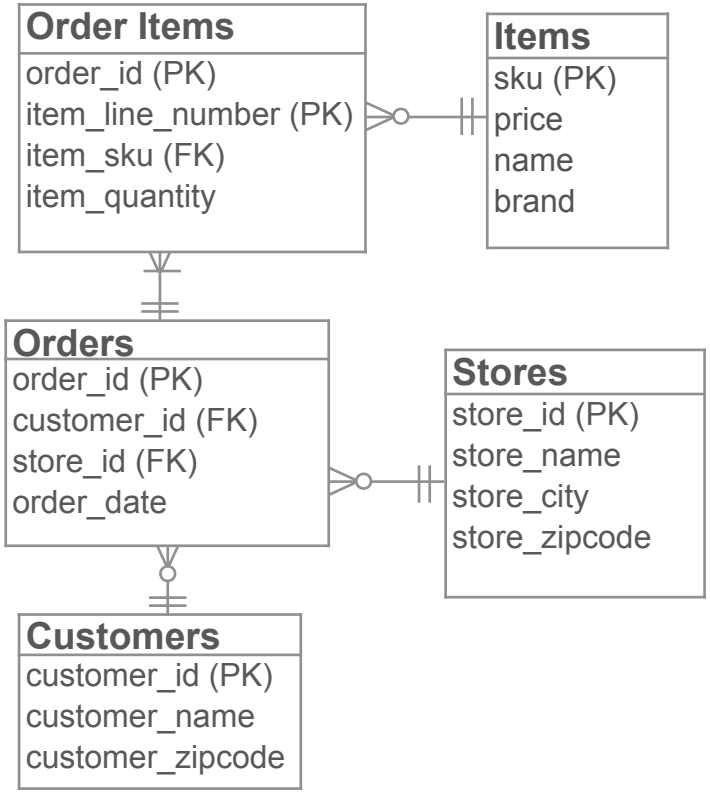
DeepLearning.AI

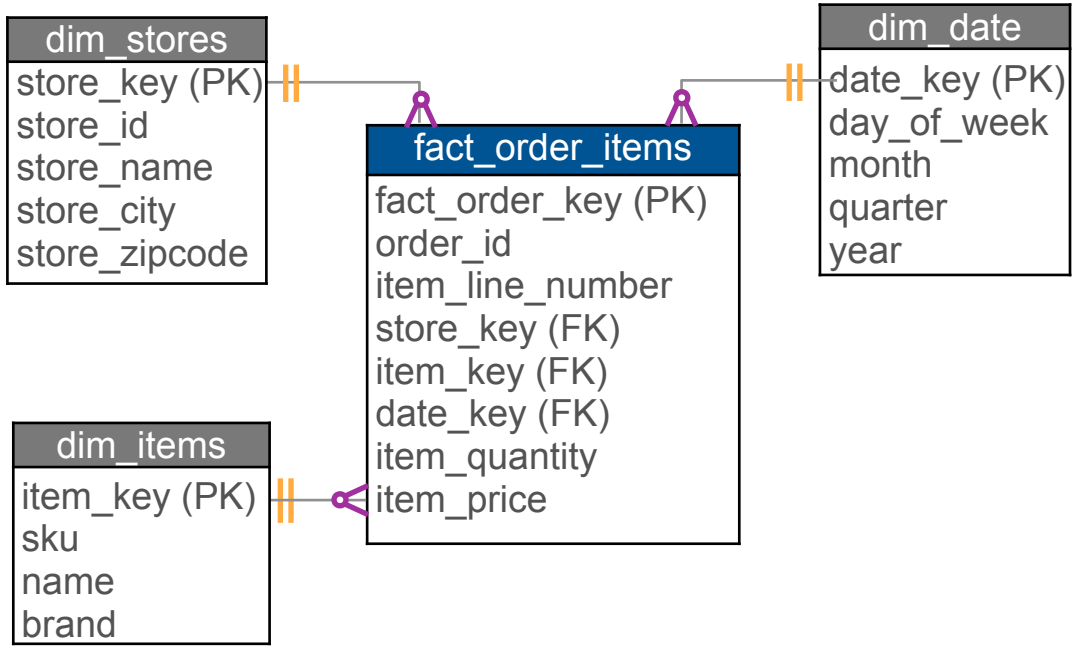
Data Modeling Techniques

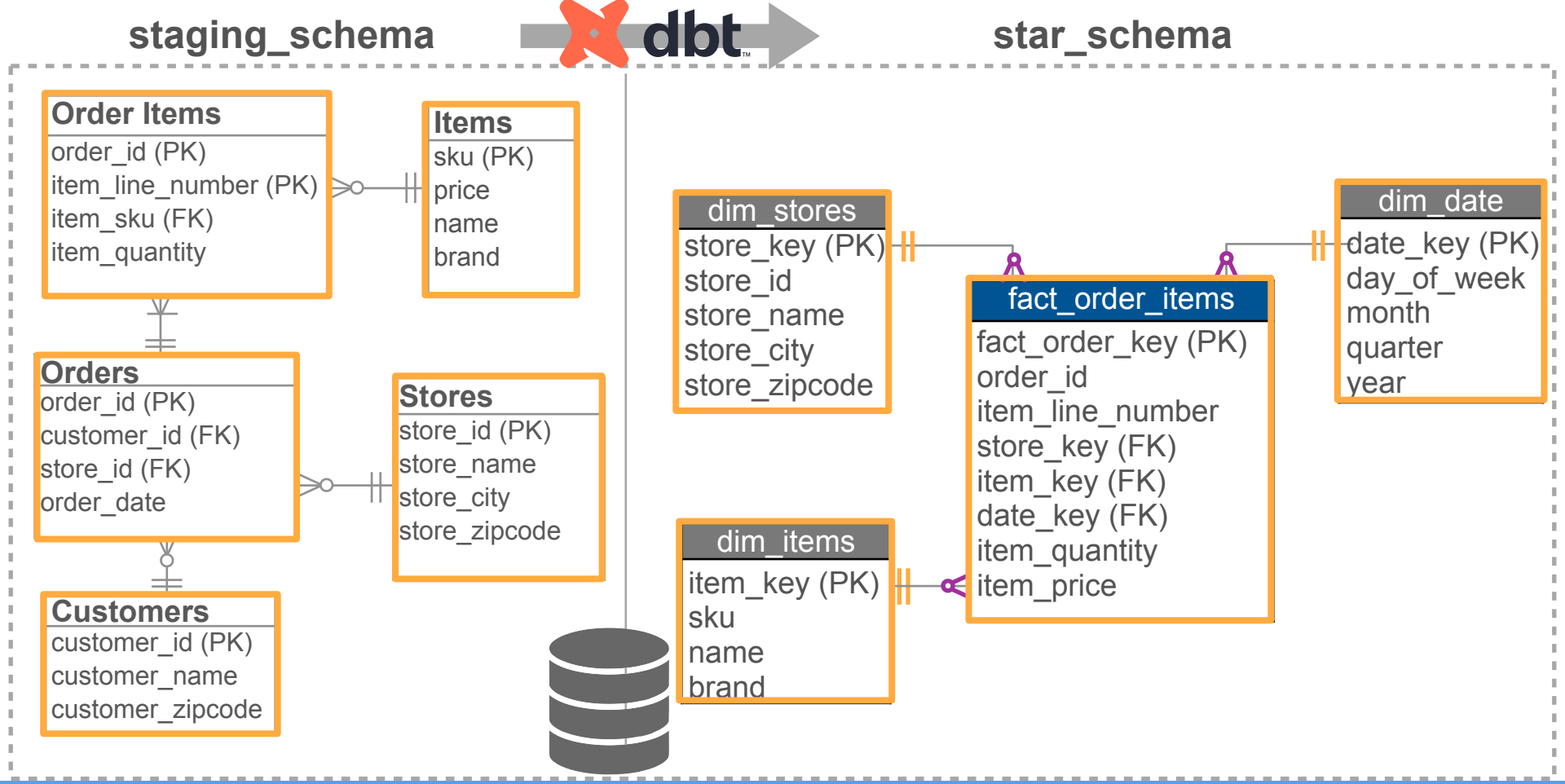
DBT Demo (Part 1)

Normalized Data

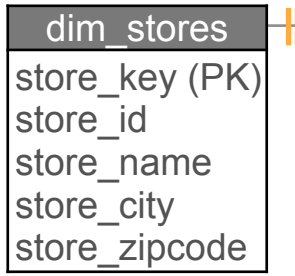
Star Schema

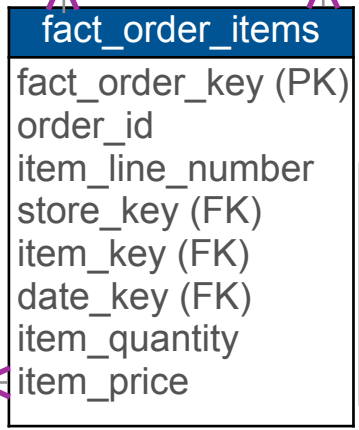


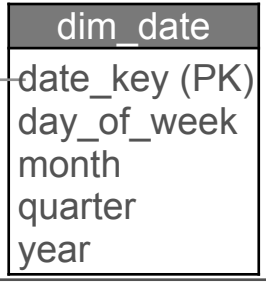



staging_schema
dbt
star_schema
Order Items
order_id (PK)
item_line_number (PK)
item_sku (FK)
item_quantity
Items
sku (PK)
price
name
brand
Orders
order_id (PK)
customer_id (FK)
store_id (FK)
order_date
Stores
store_id (PK)
store_name
store_city
store_zipcode
Customers
customer_id (PK)
customer_name
customer_zipcode
dim_stores
store_key (PK)
store_id
store_name
store_city
store_zipcode
fact_order_items
fact_order_key (PK)
order_id
item_line_number
store_key (FK)
item_key (FK)
date_key (FK)
item_quantity
item_price
dim_date
date_key (PK)
day_of_week
month
quarter
year
dim_items
item_key (PK)
sku
name
brand

```
SELECT MD5(store_id)
   as store_key,
 store_id,
 store_name,
 store_city,
 store_zipcode
FROM stores;
```

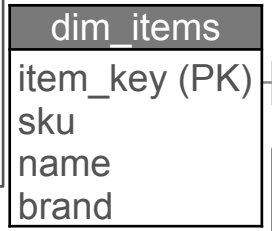







```
SELECT date_key,
 EXTRACT(DAY FROM date_key) AS day_of_week,
 EXTRACT(MONTH FROM date_key) AS month,
 EXTRACT(Quarter FROM date_key) AS quarter,
 EXTRACT(year FROM date_key) AS YEAR
FROM generate_series ('2020-01-01'::date,
 '2025-01-01'::date, '1 day'::interval) As date_key
```

```
SELECT
 MD5(sku)as item_key,
 sku,
 name,
 brand
FROM items;
```




```
Select MD5(CONCAT(OrderItems.order_id, OrderItems.item_line_number))
        AS fact_order_key,
  OrderItems.order_id, OrderItems.item_line_number,
  MD5(Orders.store_id) AS store_key, MD5(OrderItems.item_sku) AS item_key,
  Orders.order_date AS date_key, OrderItems.item_quantity,
  Items.price AS item_price
FROM OrderItems
Join Orders ON Orders.order_id = OrderItems.order_id
Join Items ON Items.sku = OrderItems.item_sku
```

dbt

- Wraps the SQL statement with a create statement
- Helps document and validate data within the data warehouse

- dbt Core:
  - open-source command line tool that you can install locally
  - communicate with your databases through adapters

- dbt Cloud:
  - runs dbt core in hosted environment with a browser-based interface

DeepLearning.AI

Data Modeling Techniques

DBT Demo (Part 2)

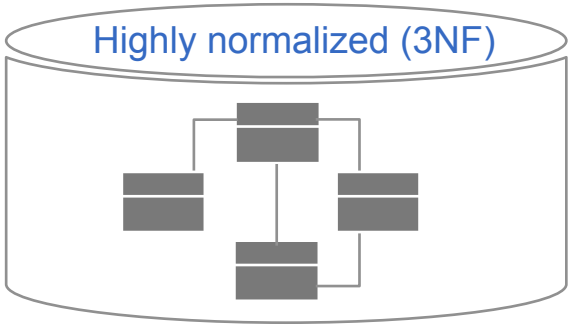
DeepLearning.AI

Data Modeling for Analytics

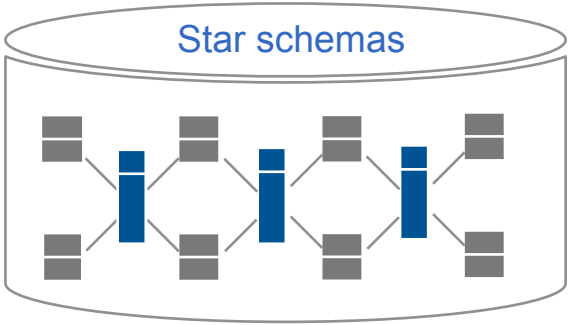
Week 1 Summary

Data Modeling Approaches

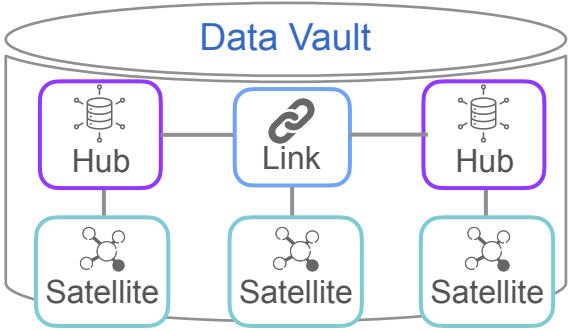
Inmon's Modeling Approach



Kimball's Modeling Approach

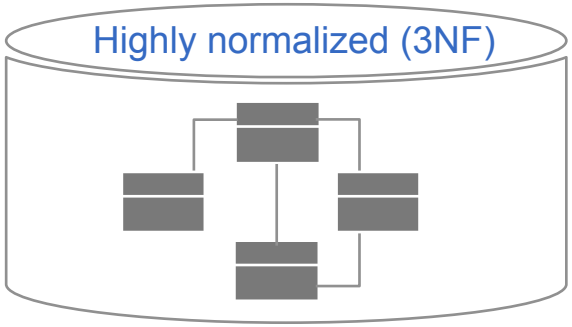


Data Vault Modeling Approach



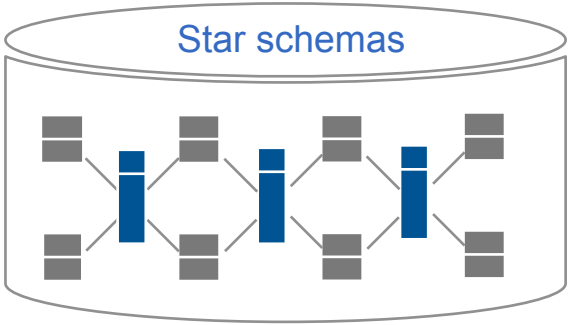
Data Modeling Approaches

Inmon's Modeling Approach

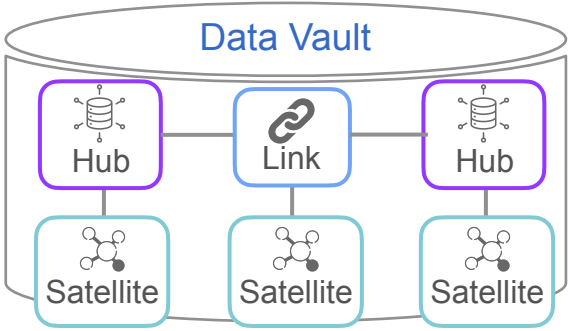


- Single source of truth for the organization
- Data is modeled in 3NF:
  - Avoid data duplication
  - Ensure the data integrity
- Analytical query:
  - Complex queries with many joins

Kimball's Modeling Approach

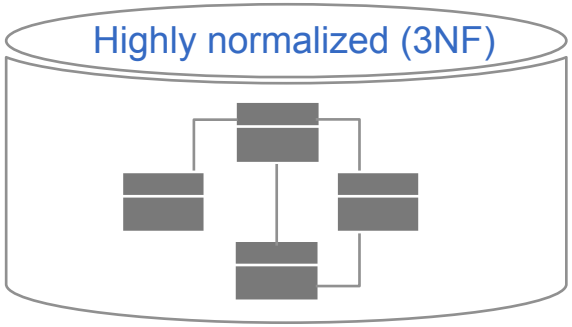


Data Vault Modeling Approach



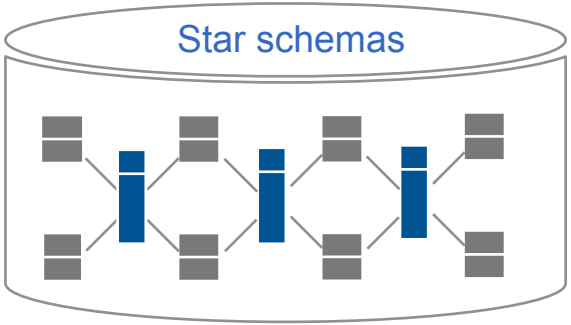
Data Modeling Approaches

Inmon's Modeling Approach



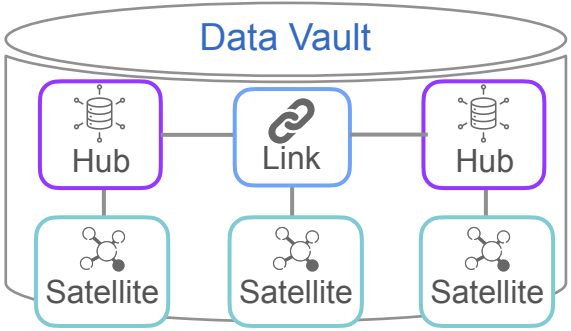
- Single source of truth for the organization
- Data is modeled in 3NF:
  - Avoid data duplication
  - Ensure the data integrity
- Analytical query:
  - Complex queries with many joins

Kimball's Modeling Approach



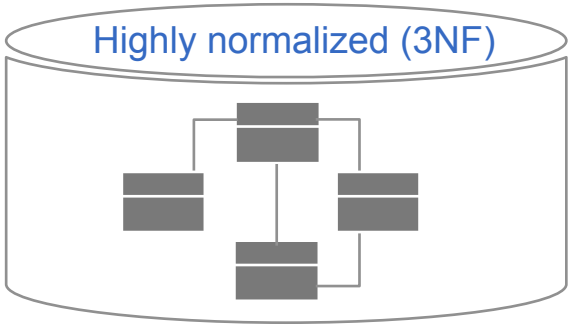
- Enables faster iteration and modeling
- Data is modeled in star schemas:
  - Fact table: business measures
  - Dimension tables: context info
- Requires good understanding of business requirements, which might not be well-defined

Data Vault Modeling Approach



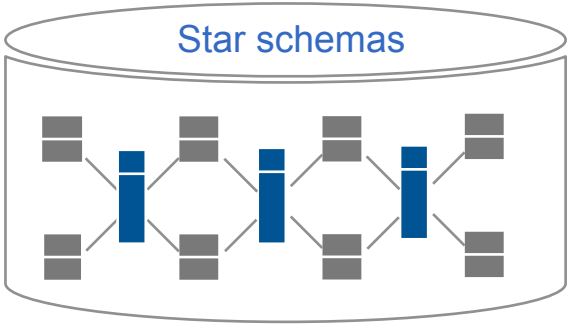
Data Modeling Approaches

Inmon's Modeling Approach



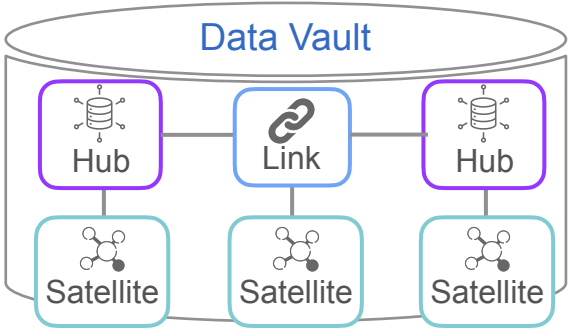
- Single source of truth for the organization
- Data is modeled in 3NF:
  - Avoid data duplication
  - Ensure the data integrity
- Analytical query:
  - Complex queries with many joins

Kimball's Modeling Approach



- Enables faster iteration and modeling
- Data is modeled in star schemas:
  - Fact table: business measures
  - Dimension tables: context info
- Requires good understanding of business requirements, which might not be well-defined

Data Vault Modeling Approach



- Offers a more flexible design to use in an agile environment
- Data vault model:
  - Hubs - core business concepts
  - Links - relationships
  - Satellite - context
- Requires downstream modeling of the data

Data Modeling Approaches

One Big Table or OBT

- No need to perform complex joins. Analytical queries are fast and simple.
- Lose the business logic in your analytics.
- The table contains duplicate information and occupies large space.