



Master of Business Administration in Digital Business & AI
2024-2026

BUSINESS ANALYTICS & DATA SCIENCE

Prof. Dr. Jan Mammen

Agenda

Business Analytics

- Introduction
- Types of Data and Databases
- Data Models
- Entity Relationship Model
- Case 1: Reverse Engineer a Database Model
- The Star and Snowflake Schema
- Analyzing Data with Power BI
- Case 2: Analyzing Procurement Transactions

Data Science

Agenda

Business Analytics

Introduction

Types of Data and Databases

Data Models

Entity Relationship Model

Case 1: Reverse Engineer a Database Model

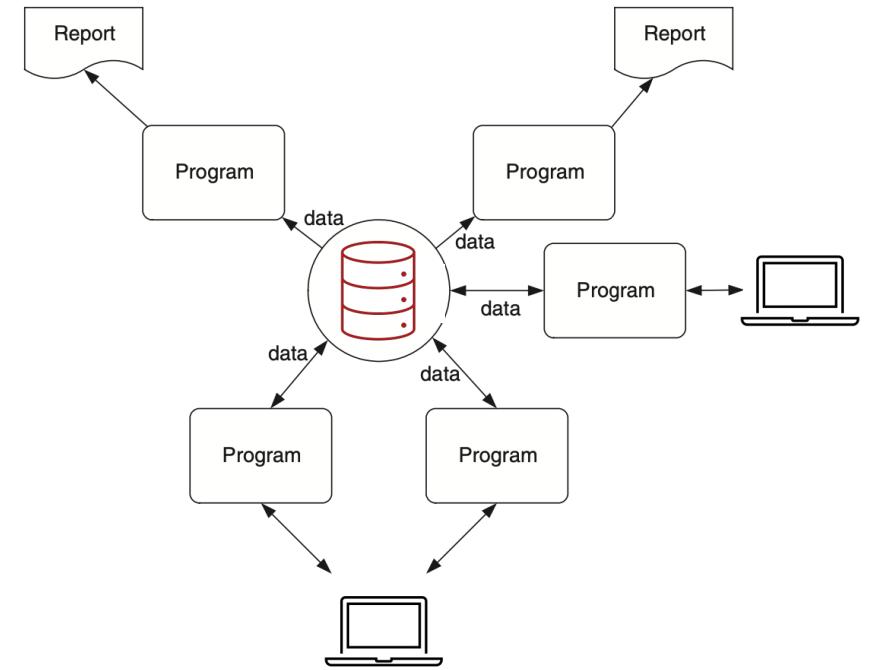
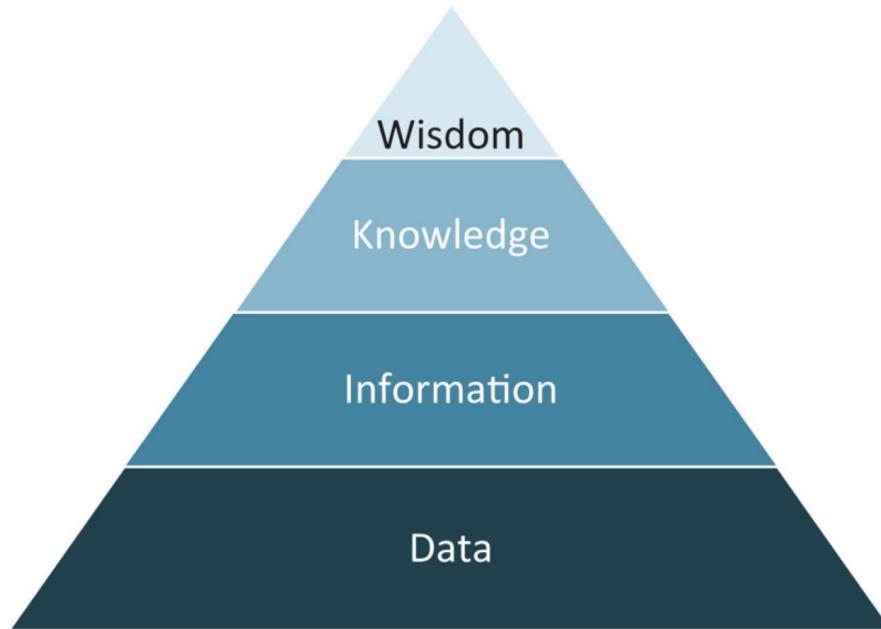
The Star and Snowflake Schema

Analyzing Data with Power BI

Case 2: Analyzing Procurement Transactions

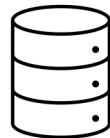
Data Science

Every Information System Is Built on Data



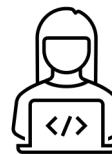
Why Efficient Data Management Is Important

Data Sources



Steadily growing
sources of data

Advanced Analytics



Non-domain experts such as
data scientist need to be able to
work efficiently with data (e.g.,
data catalogues)

Customer experience



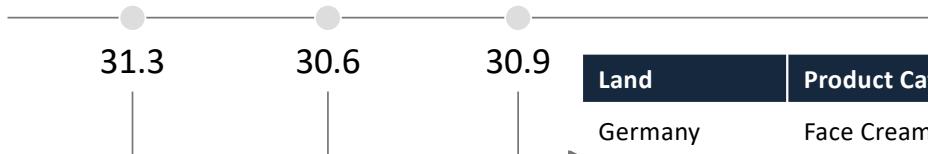
Amazon found for every
100 milliseconds of
latency; they lost 1%
revenue.*

*<https://glinden.blogspot.com/2006/11/marissa-mayer-at-web-20.html>

Why Data Management Has Become More Challenging - Corporate Planning Process

Sales planning

Forecast 1 Forecast 2 Forecast 3



Land	Product Category	Sales (kEUR)
Germany	Face Cream	100,000
France	Face Cream	50,000
...

Production Planning

Site	Product Category	Quantity
Hamburg	Face Cream	10.000
Leipzig	Face Cream	2.000

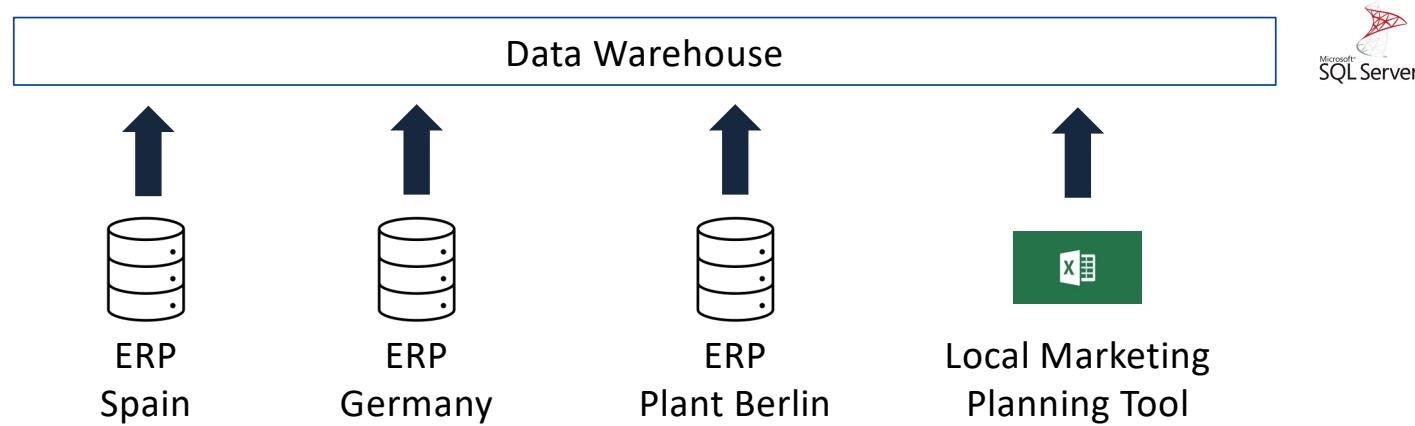
Purchase Planning

Material	Supplier	ML
Glycerin	Neutro GmbH	50.000
Panthenol	Chemie AG	85.000

Why Data Management Has Become More Challenging - Corporate Planning Process

Type of data	Source System
1 Forecasted sales quantity existing products	
2 Forecasted sales quantity new products	
3 Prices	
4 Marketing Budget	

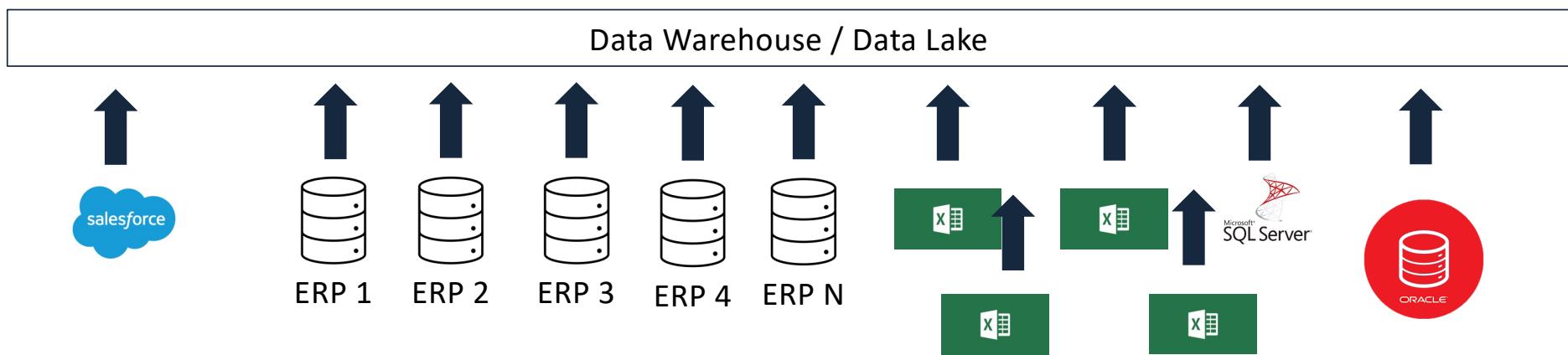
Why Data Management Has Become More Challenging - Corporate Planning Process



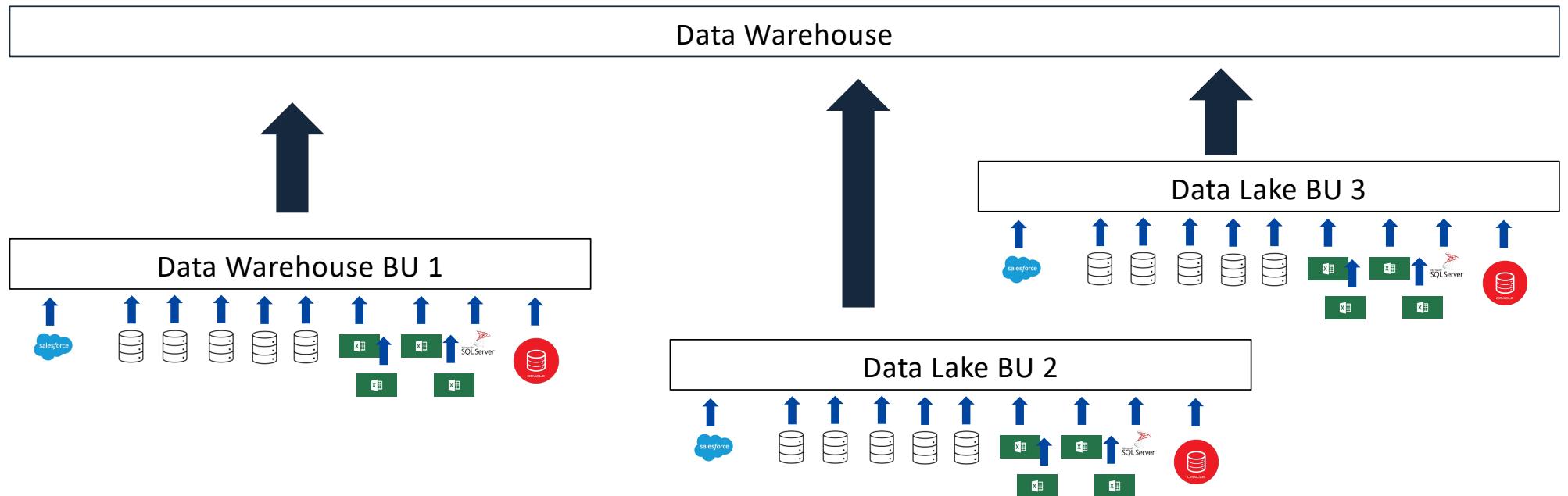
Why Data Management Has Become More Challenging - Corporate Planning Process

Type of data	Source System
1 Forecasted sales quantity existing products	
2 Forecasted sales quantity new products	
3 Prices	
4 Marketing Budget	

Why Data Management Has Become More Challenging - Corporate Planning Process



Why Data Management Has Become More Challenging - Corporate Planning Process





*How many ERP-Systems is Siemens AG
using?*

*How many PB of data are saved in their core
platform*

Drivers of Complex Data Landscapes

Technical and Architectural Factors

- **Legacy Systems:** Incompatible formats, missing APIs, outdated technology.
- **Hybrid IT (Cloud + On-Prem):** Data is spread across environments with different integration capabilities.
- **Tool Proliferation:** Overuse of specialized tools leads to fragmented data without a shared model.
- **Vendor Lock-in:** Proprietary systems discourage data portability and integration.

Organizational and Operational Factors

- **Departmental Autonomy ("Shadow IT"):** Teams independently adopt tools, creating silos.
- **Mergers & Acquisitions:** Inherited systems and inconsistent architectures.
- **Lack of Central Data Strategy:** Growth and tool adoption without enterprise-wide alignment.
- **Cultural Silos:** Reluctance to share or harmonize data across business units.

Advanced Analytics as a Driver for a Complex Data Landscapes

Large number of sources



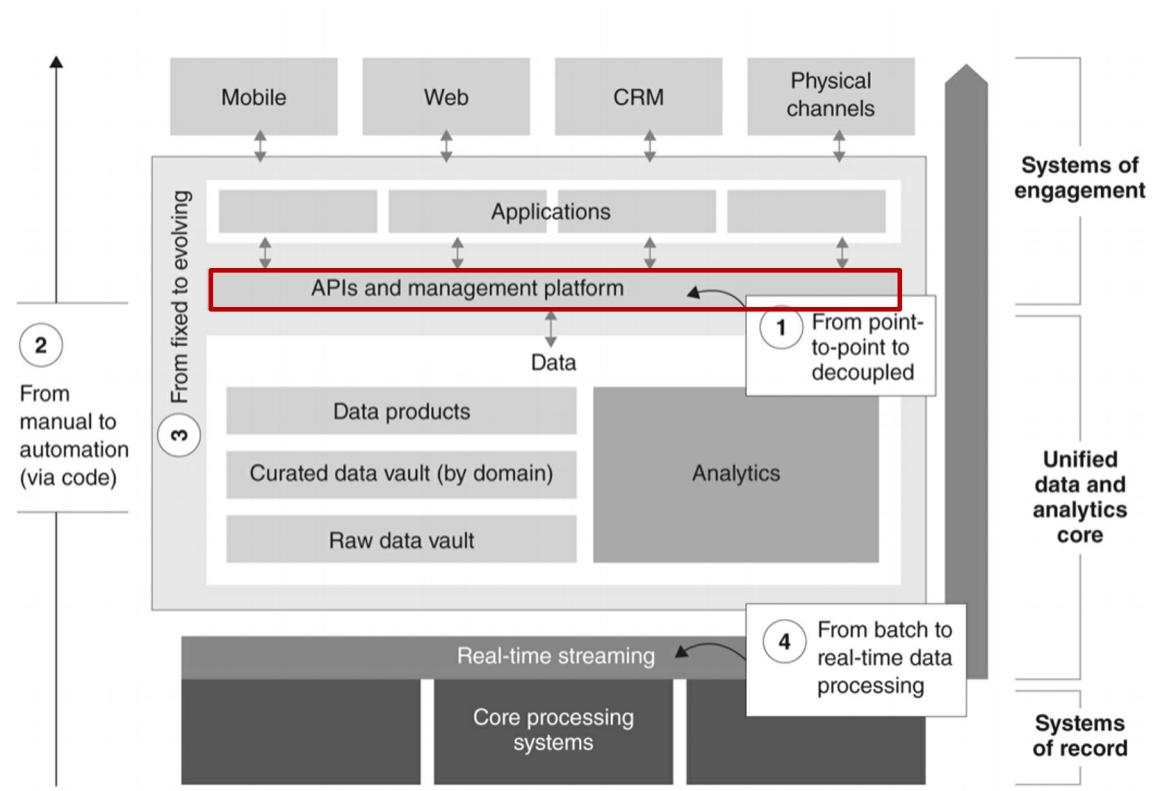
Each use case requires different data

Scattered Data



Data is often copied and scattered in different applications

Given the Number of Systems Companies Try to Get Beyond Point-to-point Communication



Source: Lamarre, E., Smaje, K., & Zemmel, R. (2023). Rewired: the McKinsey guide to outcompeting in the age of digital and AI.

Evolving Data Landscape of Data Platforms

	Data Warehouse	Data Lake	Data Fabric	Data Mesh	
Architecture	<ul style="list-style-type: none"> Central Focus on modeling and analysis 	<ul style="list-style-type: none"> Central Focus on scalability 	<ul style="list-style-type: none"> Central layers on top of distributed data Focus on data management, integration and access 	<ul style="list-style-type: none"> Decentralized & distributed Focus on product thinking Focus on organizational change 	
Data	Format	<ul style="list-style-type: none"> Primarily structured format 	<ul style="list-style-type: none"> Structured format Unstructured format 	<ul style="list-style-type: none"> Any format 	<ul style="list-style-type: none"> Data as a product Any format
	Status	<ul style="list-style-type: none"> Mostly pre-processed Consolidated 	<ul style="list-style-type: none"> Mostly Raw 	<ul style="list-style-type: none"> Any status 	<ul style="list-style-type: none"> According to data product How customer needs it
	Types	<ul style="list-style-type: none"> Primarily internal data Mostly ERP / CRM / PLM 	<ul style="list-style-type: none"> Internal and external data Various data sources 	<ul style="list-style-type: none"> Internal and external data Various data sources 	<ul style="list-style-type: none"> Internal and external data Various data sources
	Sourcing & Storage	<ul style="list-style-type: none"> Data replication (ETL) Data is centrally stored (in the data warehouse) 	<ul style="list-style-type: none"> Data replication (ELT) Data is centrally stored (in the data lake) 	<ul style="list-style-type: none"> Either data replication (ETL or ELT) with central data storage (in the data fabric) Or data virtualization with decentral data storage (original data sources) 	<ul style="list-style-type: none"> Data is stored decentrally (original data sources) but are logically integrated into the data mesh
	Access	<ul style="list-style-type: none"> Central access 	<ul style="list-style-type: none"> Central access 	<ul style="list-style-type: none"> Central access (self-service) through fabric (API) layer, using data catalog or knowledge graph data model 	<ul style="list-style-type: none"> Decentral access (self-service) where data is produced
	Ownership & Governance	<ul style="list-style-type: none"> Central 	<ul style="list-style-type: none"> Central 	<ul style="list-style-type: none"> Central (empowered by strong metadata foundation) 	<ul style="list-style-type: none"> Decentral (empowered by domain or product owners)

Data Mesh

Data ownership by domain

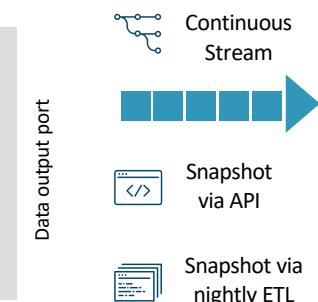
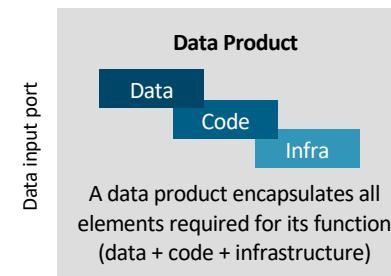
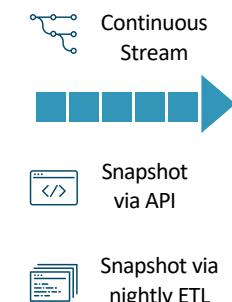
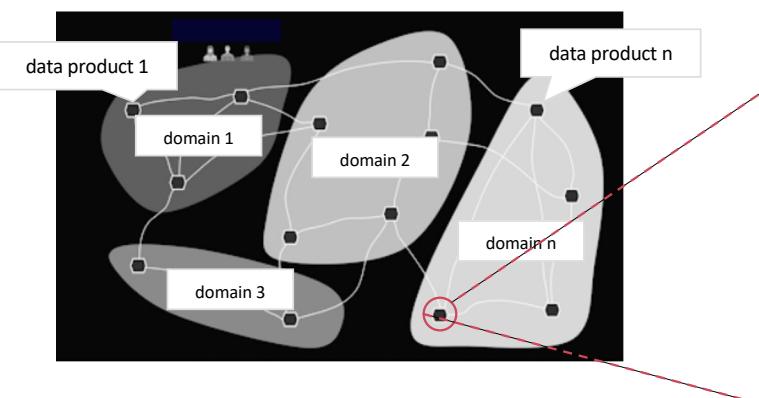
- Data "lives" where it's created and is owned by those who understand its structure and value the best
- Domain-driven decentralization of responsibility for data
- Data is governed where it is created

Data as a product

- Data owners think product-oriented and customer-centric
- The data owner is responsible for all aspects of the data (e.g., quality, form, pre-processing)

Data available everywhere

- Data is locally produced and made available in a decentral way (e.g. via event streaming), to be consumed by other teams in the company
- From everywhere in the company, all data is accessible via self-service



Agenda

Business Analytics

Introduction

Types of data and databases

Data models

Entity relationship model

Case 1: Reverse engineer a database model

The Star and Snowflake Schema

Analyzing Data with Power BI

Case 2: Analyzing procurement transactions

Data Science

Agenda

Business Analytics

Introduction

Types of Data and Databases

Data Models

Entity Relationship Model

Case 1: Reverse Engineer a Database Model

The Star and Snowflake Schema

Analyzing Data with Power BI

Case 2: Analyzing Procurement Transactions

Data Science

Tabular Data Can Be Saved in Various Formats

	A	B	C
1	name	price	volume
2	a	10.12	20
3	b	90.5	100
4	c	8.99	40

Tabular Data Can Be Saved in Various Formats

CSV

```
name, price, volume
a, 10.12, 20
b, 90.5, 100
c, 8.99, 40
```

XML

```
<?xml version='1.0' encoding='utf-8'?>
<data>
  <row>
    <index>1</index>
    <name>a</name>
    <price>10.12</price>
    <volume>20</volume>
  </row>
  <row>
    <index>2</index>
    <name>b</name>
    <price>90.5</price>
    <volume>100</volume>
  </row>
  <row>
    <index>3</index>
    <name>c</name>
    <price>8.99</price>
    <volume>40</volume>
  </row>
</data>
```

JSON

```
{
  "1": {"name": "a", "price": 10.12, "volume": 20},
  "2": {"name": "b", "price": 90.5, "volume": 100},
  "3": {"name": "c", "price": 8.99, "volume": 40}
}
```

Example for Encoding an Image



The "Grinning Face" emoji, from the Twemoji set

Base64 encoded image

iVBORw ➔ 10001001010100000100111001000111

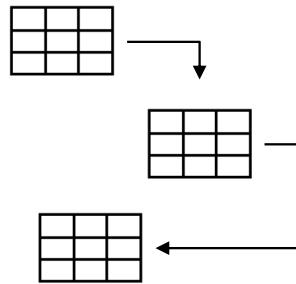
Types of Database

Key-Value Database

Key	Value
Paul	87077
Mike	8179

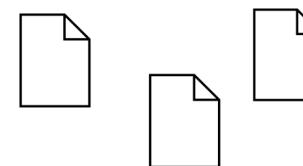
Redis,
DynamoDB

Relational Databases



SQL-Server, Postgres,
MySQL,

Document Databases



MongoDB, CouchDB,
CosmosDB

Graph Databases



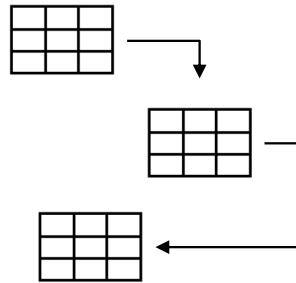
Neo4j,
Amazon Neptune

Types of Database

Key-Value Database	
Key	Value
Paul	87077
Mike	8179

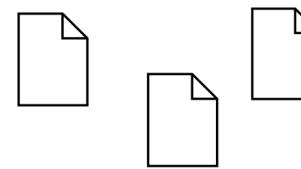
Redis,
DynamoDB

Relational Databases



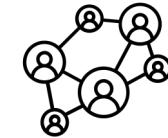
SQL-Server, Postgres,
MySQL,

Document Databases



MongoDB, CouchDB,
CosmosDB

Graph Databases



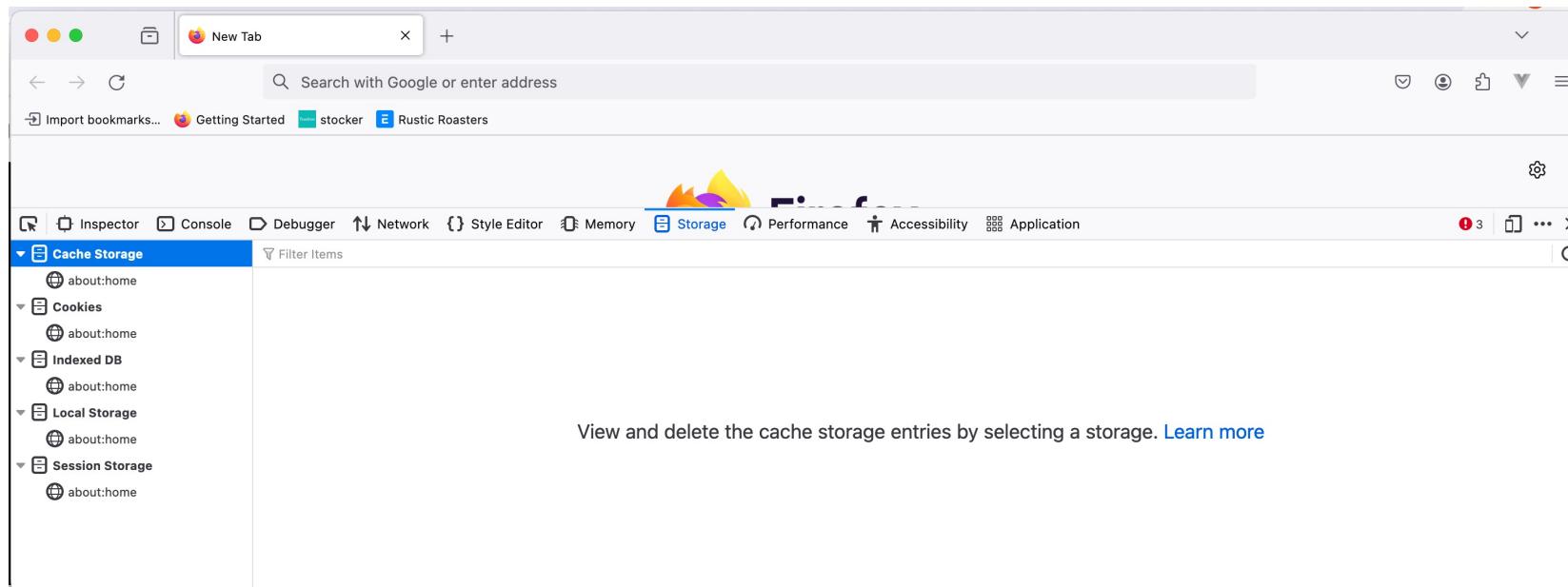
Neo4j,
Amazon Neptune

Key-value Store – Simplest Version of Database

Key	Value
Steven	001 - 4797797
Maria	0030 - 2312321

```
product_prices = {  
    "Apple": 0.5,  
    "Banana": 0.25,  
    "Cherry": 0.75,  
    "Date": 1.0  
}
```

Key-value Store – Simplest Version of Database



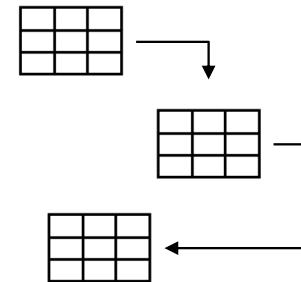
Types of Database

Key-Value Database

Key	Value
Paul	87077
Mike	8179

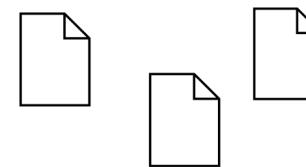
Redis,
DynamoDB

Relational Databases



SQL-Server, Postgres,
MySQL,

Document Databases



MongoDB, CouchDB,
CosmosDB

Graph Databases



Neo4j,
Amazon Neptune

Example for a Linked in Profile with a Relational Database

<http://www.linkedin.com/in/williamhgates>



Bill Gates
Greater Seattle Area | Philanthropy

Summary
Co-chair of the Bill & Melinda Gates Foundation. Chairman, Microsoft Corporation. Voracious reader. Avid traveler. Active blogger.

Experience
Co-chair • Bill & Melinda Gates Foundation
2000 – Present
Co-founder, Chairman • Microsoft
1975 – Present

Education
Harvard University
1973 – 1975
Lakeside School, Seattle

Contact Info
Blog: thegatesnotes.com
Twitter: @BillGates

User_id	First_name	Last_name	Summary	region_id	Industry_id
251	Bill	Gates	Co-chair of ..	us:91	131

id	region_name
us:7	Greater Boston Area
us:91	Greater Seattle Area

id	industry_name
43	Financial Services
48	Construction
131	Philanthropy

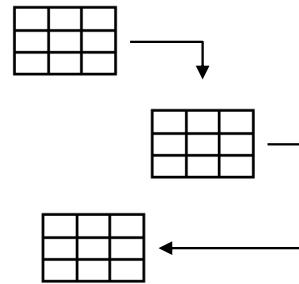
Types of Database

Key-Value Database

Key	Value
Paul	87077
Mike	8179

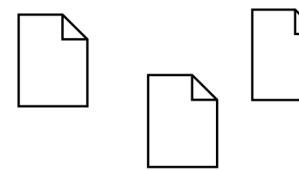
Redis,
DynamoDB

Relational Databases



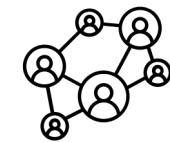
SQL-Server, Postgres,
MySQL,

Document Databases



MongoDB, CouchDB,
CosmosDB

Graph Databases



Neo4j,
Amazon Neptune

Example for a Linked in Profile with a Document Database

<http://www.linkedin.com/in/williamhgates>



Bill Gates
Greater Seattle Area | Philanthropy

Summary
Co-chair of the Bill & Melinda Gates Foundation. Chairman, Microsoft Corporation. Voracious reader. Avid traveler. Active blogger.

Experience
Co-chair • Bill & Melinda Gates Foundation
2000 – Present
Co-founder, Chairman • Microsoft
1975 – Present

Education
Harvard University
1973 – 1975
Lakeside School, Seattle

Contact Info
Blog: [thegatesnotes.com](https://www.gatesnotes.com/)
Twitter: @BillGates

```
{
  "user_id": 251,
  "first_name": "Bill",
  "last_name": "Gates",
  "summary": "Co-chair of the Bill & Melinda Gates... Active blogger.",
  "region_id": "us:91",
  "industry_id": 131,
  "photo_url": "/p/7/000/253/05b/308dd6e.jpg",
  "positions": [
    {"job_title": "Co-chair", "organization": "Bill & Melinda Gates Foundation"},
    {"job_title": "Co-founder, Chairman", "organization": "Microsoft"}
  ],
  "education": [
    {"school_name": "Harvard University", "start": 1973, "end": 1975},
    {"school_name": "Lakeside School, Seattle", "start": null, "end": null}
  ],
  "contact_info": {
    "blog": "https://www.gatesnotes.com/",
    "twitter": "https://twitter.com/BillGates"
  }
}
```

The Document Structure Is Very Easy to Handle with Object-oriented Programming: Example Mapping a Documents to a Python Class

```
class Customer:  
    def __init__(self, name, balance,  
address):  
        self.name = name  
        self.balance = balance  
        self.address = address  
  
    {  
        "name": "Isabell Wang",  
        "balance": 200,  
        "address": {  
            "city": "San Francisco",  
            "state": "CA",  
            "zip": 94107,  
            "street": "123 Main St"  
    }
```

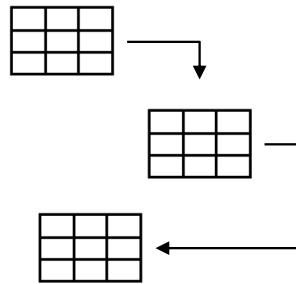
Types of Database

Key-Value Database

Key	Value
Paul	87077
Mike	8179

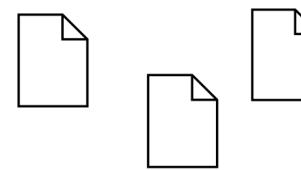
Redis,
DynamoDB

Relational Databases



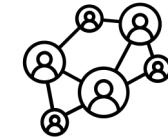
SQL-Server, Postgres,
MySQL,

Document Databases



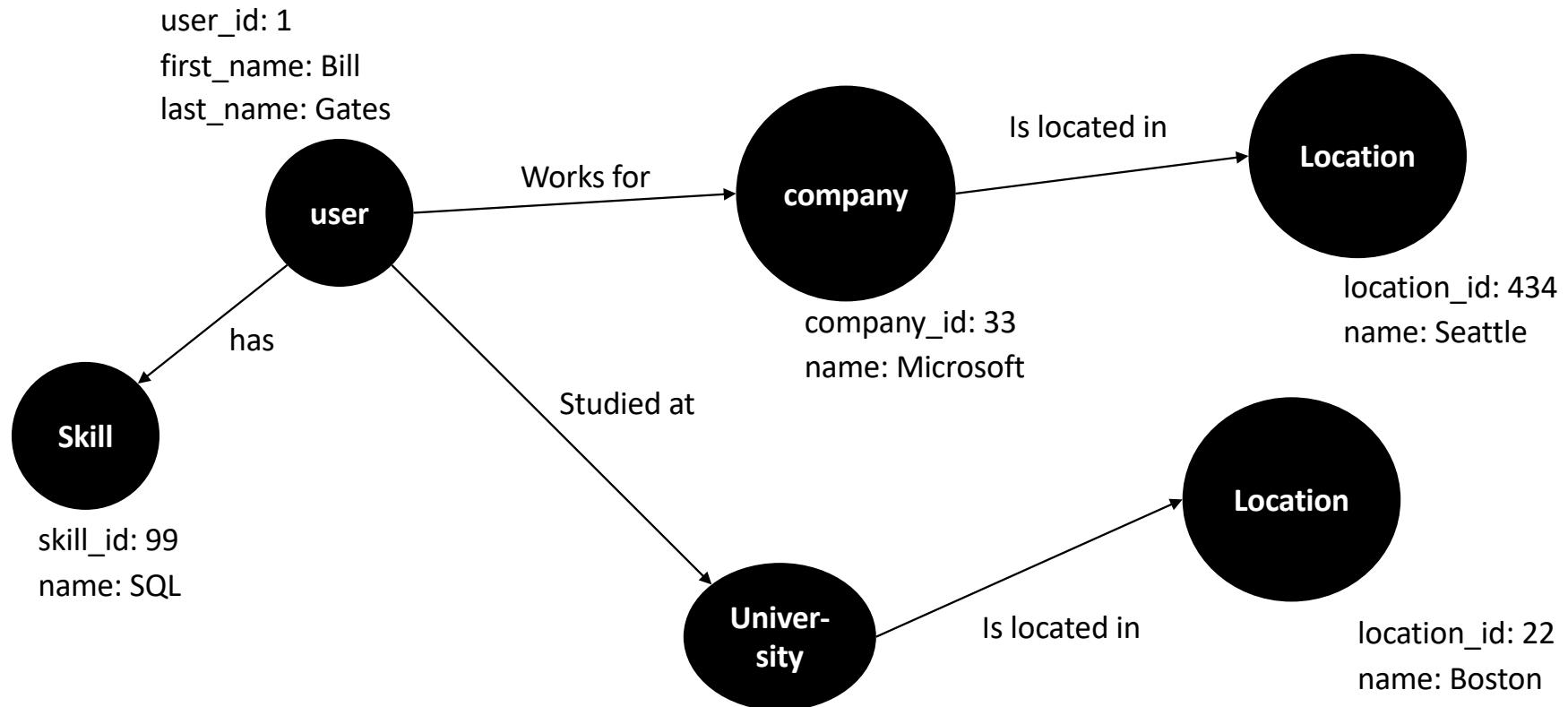
MongoDB, CouchDB,
CosmosDB

Graph Databases



Neo4j,
Amazon Neptune

Example for a Graph Database Model



Each Database Type Has Different Strength and Weaknesses

Database Type	Pros	Cons	Use Cases
Key-value	Simple, fast, and scalable	Limited query capabilities and data relationships	Real-time analytics, content management, caching, session management
Relational	Data integrity and complex queries	Schema rigidity and scalability challenges	Financial systems, inventory management, Transaction-based systems
Document	Schema flexibility and rich data structures	Data duplication and consistency issues	Content management, IoT, blogging, web analytics, mobile applications
Graph	Data relationships and traversal performance	Data size and complexity limitations	Fraud detection, social networks, route planning

Agenda

Business Analytics

Introduction

Types of Data and Databases

Data Models

Entity Relationship Model

Case 1: Reverse Engineer a Database Model

The Star and Snowflake Schema

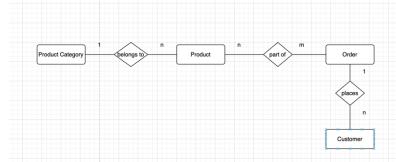
Analyzing Data with Power BI

Case 2: Analyzing Procurement Transactions

Data Science

What Is a Data Model

Conceptual data model



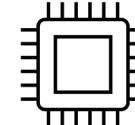
- Technology-independent specification of the data to be held in the database
- Used for communication between the data modeler and business stakeholders

Logical data model

	sales_order	sales_order_items	customer
□	name text	□	name text
□	order_date date	□	customer_type text
□	requested_date date	□	customer_group text
□	delivery_date date	□	territory text
□	customer text	□	parent text
□	warehouse text	□	delivery_date date
□	product_id text	□	requested_date date
□	quantity real		

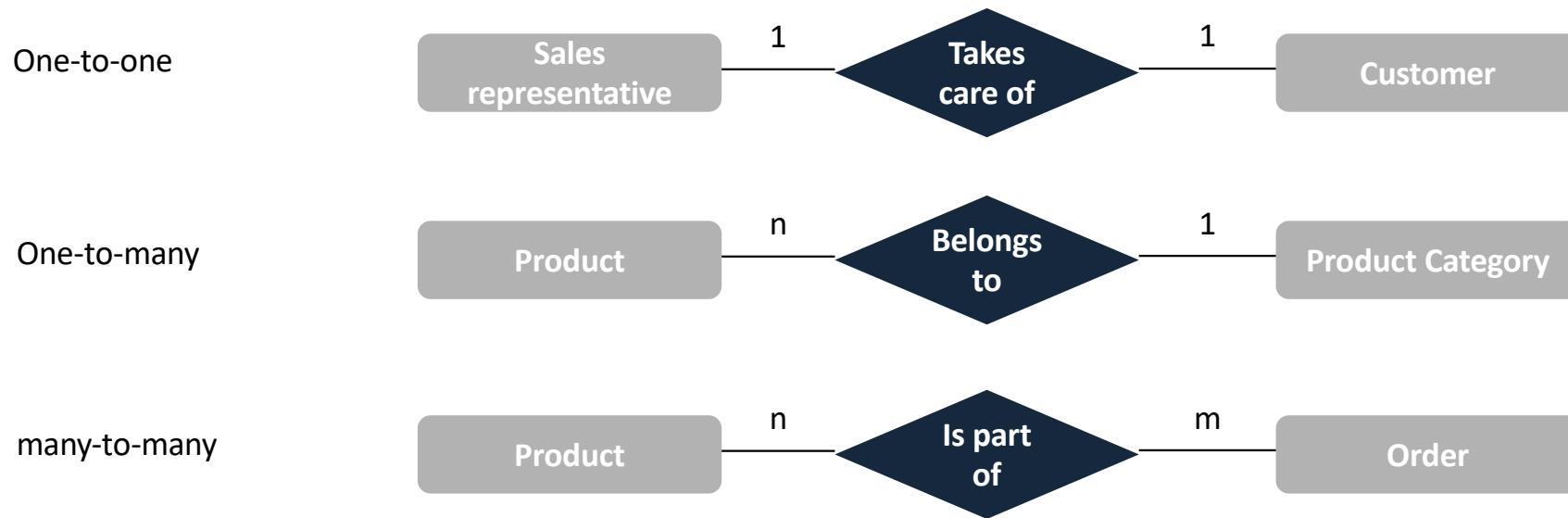
- Translation of the conceptual model into structures that can be implemented using a database management system (DBMS)
- Typically specifies tables and columns

Physical data model

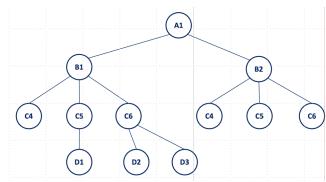


- Incorporates any changes necessary to achieve adequate performance and is also presented in terms of tables and columns
- Together with a specification of physical storage and access mechanisms

A Conceptual Database Model Essentially Defines Relationships



Conceptual Data Models



Hierarchical Data Model



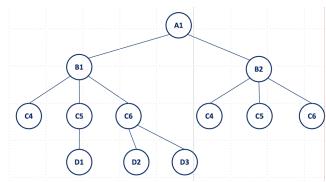
Network Data Model

order_id	product_id	customer_id	amount	date	...
251	2	55	100	2024-02-12	...
id	name	weight	color
1	shorts	20	green
2	shoe	550	black
...

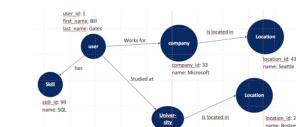
id	name	city	Email	...
1	Wang	Beijing	wang@...	...
...
55	Smith	New York	smith@...	...

Relational Data Model

Conceptual Data Models



Hierarchical Data Model



Network Data Model

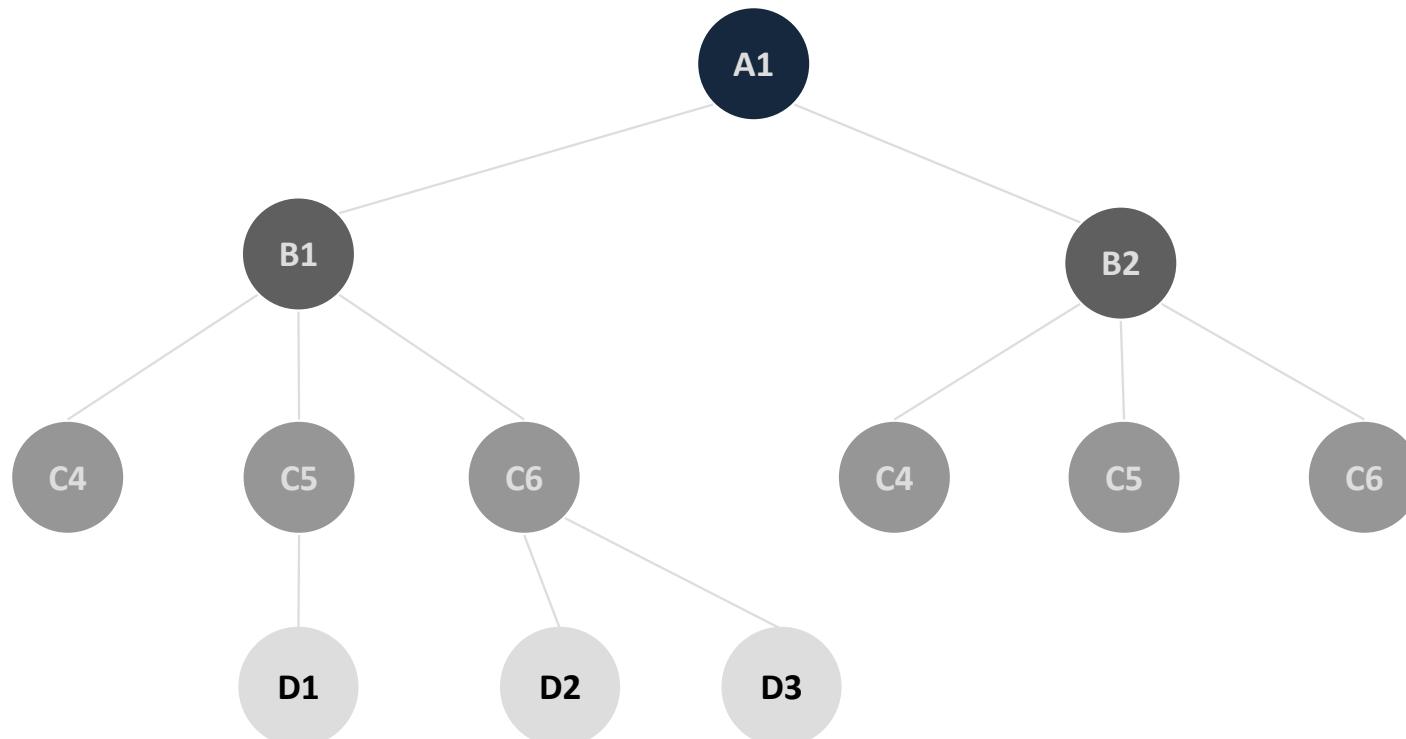
order_id	product_id	customer_id	amount	date	...
251	2	55	100	2024-02-12	...
...

id	name	weight	color	...
1	shorts	20	green	...
2	shoe	550	black	...
...

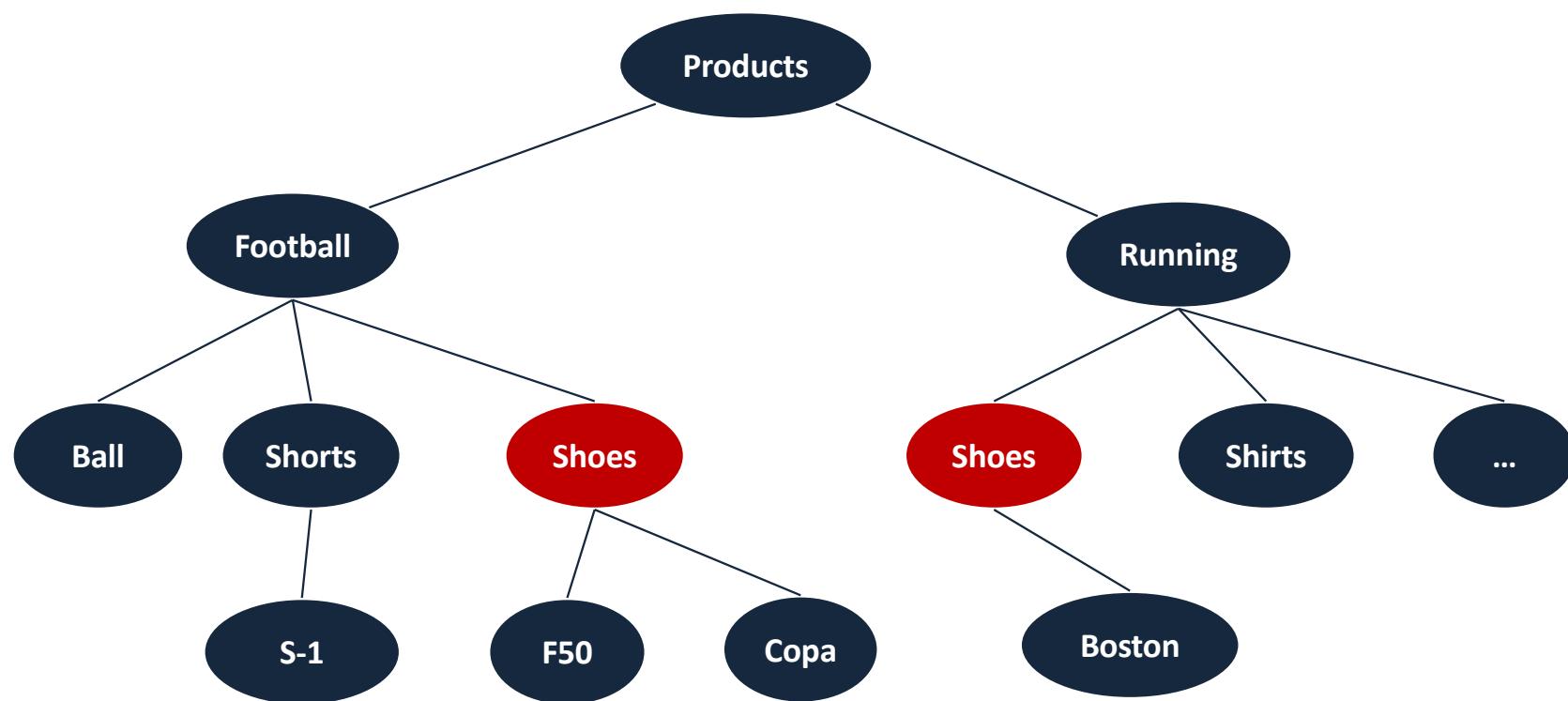
id	name	city	Email	...
1	Wang	Beijing	wang@...	...
...
55	Smith	New York	smith@...	...

Relational Data Model

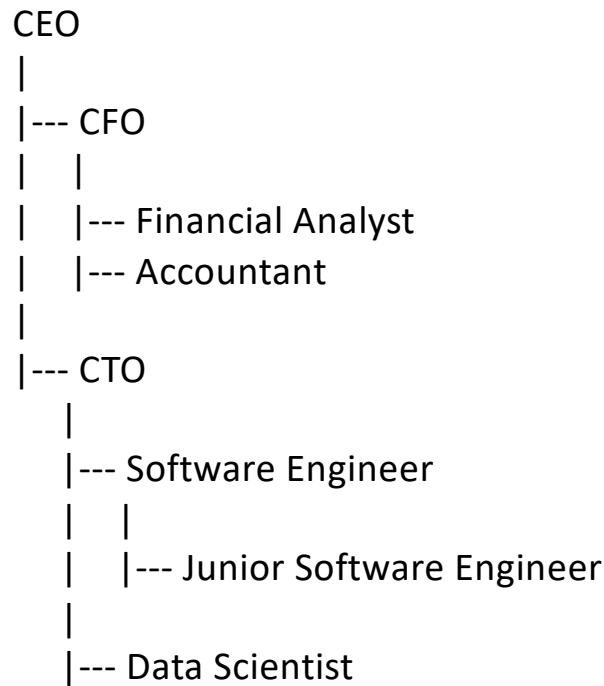
Hierarchical Database Model



Hierarchical Database Model



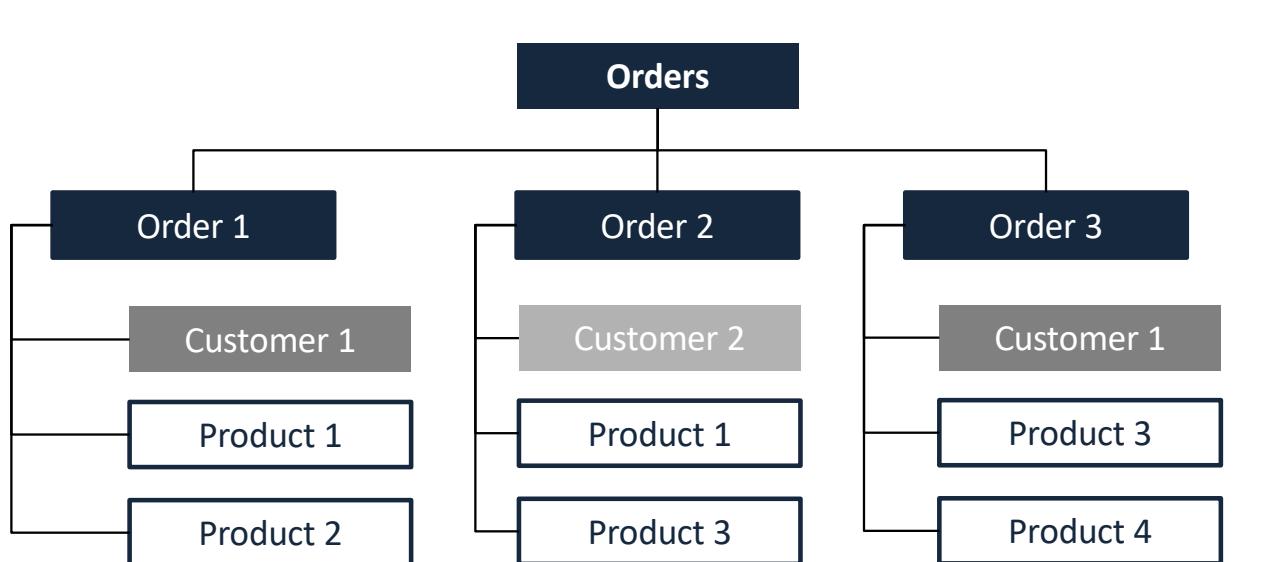
The Windows System Registry Follows a Hierarchical Model



Local File System Can Be Considered as a Hierarchical Database

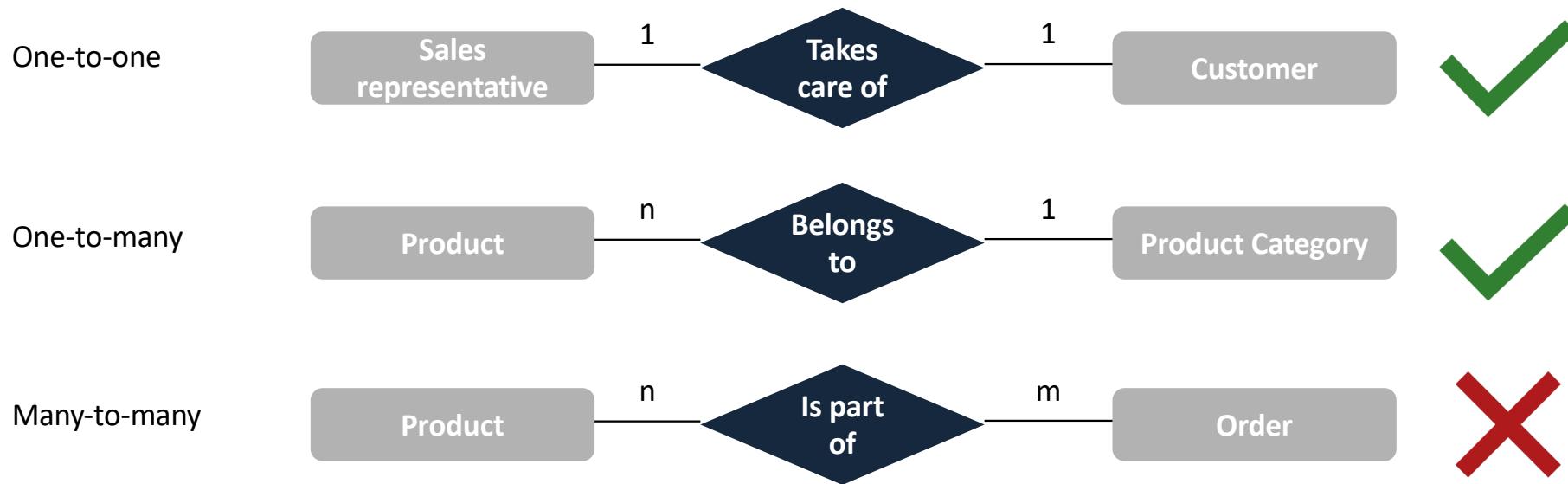
```
C:\  
|--- Program Files  
|   |  
|   |--- Microsoft Office  
|   |--- Google  
|   |   |  
|   |   |--- Chrome  
|  
|--- Users  
|   |  
|   |--- User1  
|   |   |  
|   |   |--- Documents  
|   |   |--- Pictures  
|  
|--- User2  
|   |  
|   |--- Documents  
|   |--- Pictures
```

Limitations of the Hierarchical Database Model

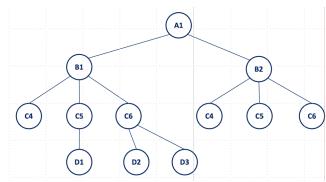


- High data redundancy
- Complex queries

Many-to-many Relationships Can Not Be Handled Well with Hierarchical Data Models



Conceptual Data Models



Hierarchical Data Model



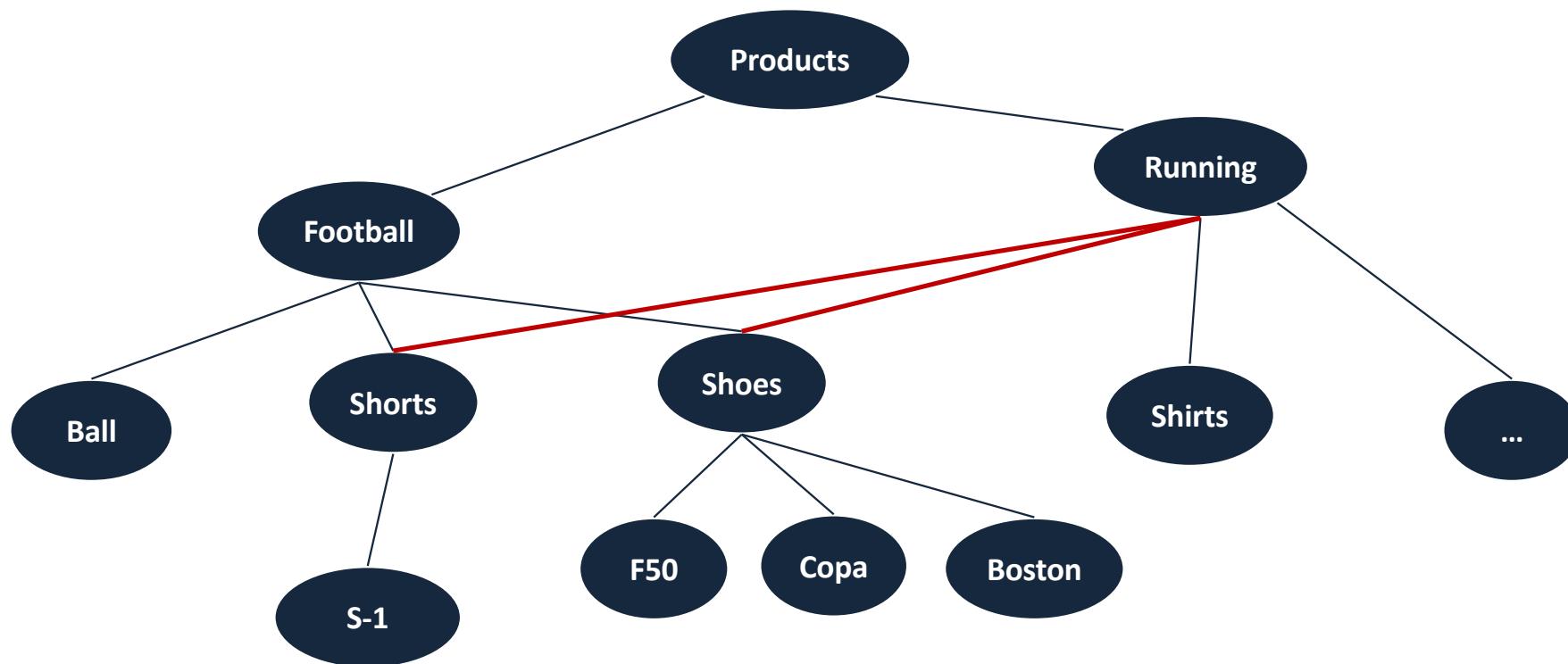
Network Data Model

order_id	product_id	customer_id	amount	date	...
251	2	55	100	2024-02-12	...
id	name	weight	color
1	shorts	20	green
2	shoe	550	black
...

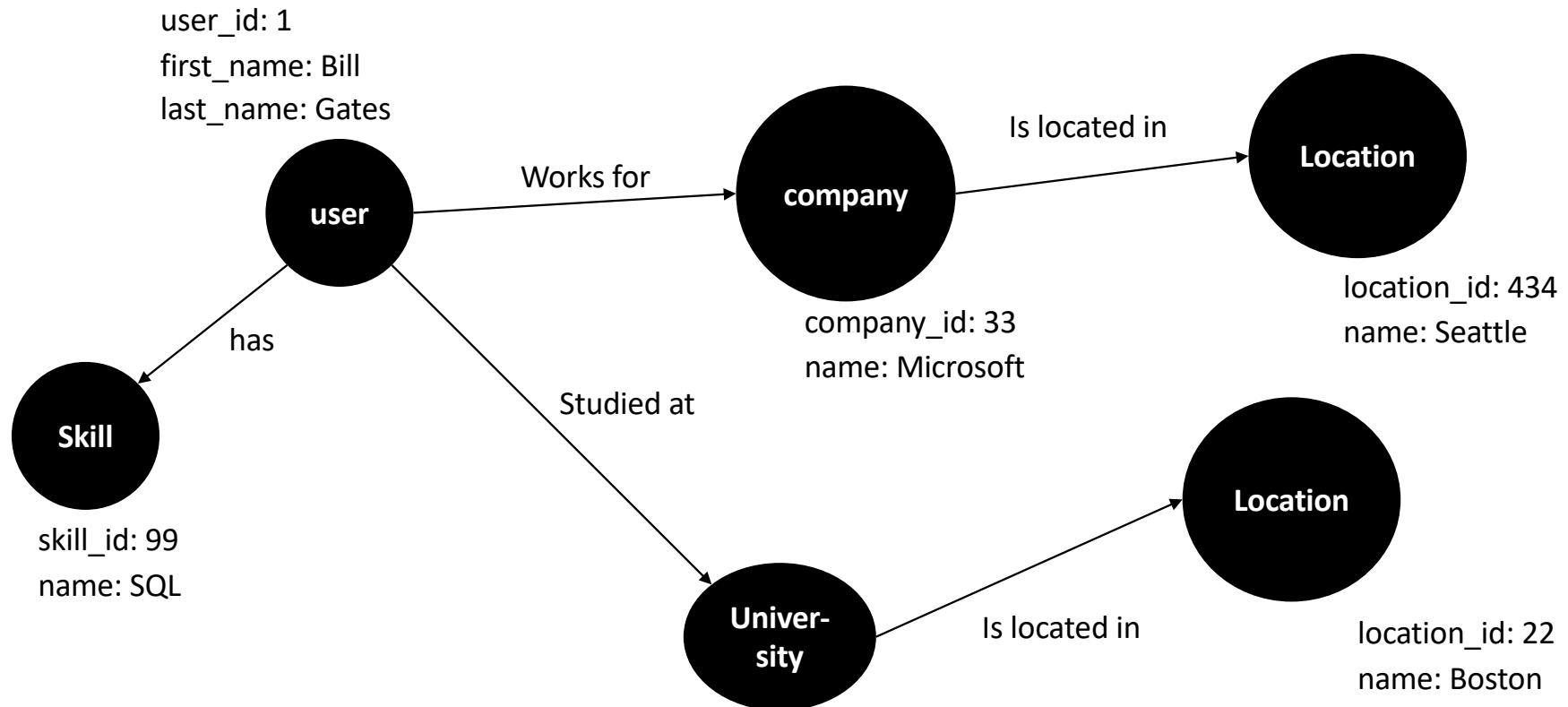
id	name	city	Email	...
1	Wang	Beijing	wang@...	...
...
55	Smith	New York	smith@...	...

Relational Data Model

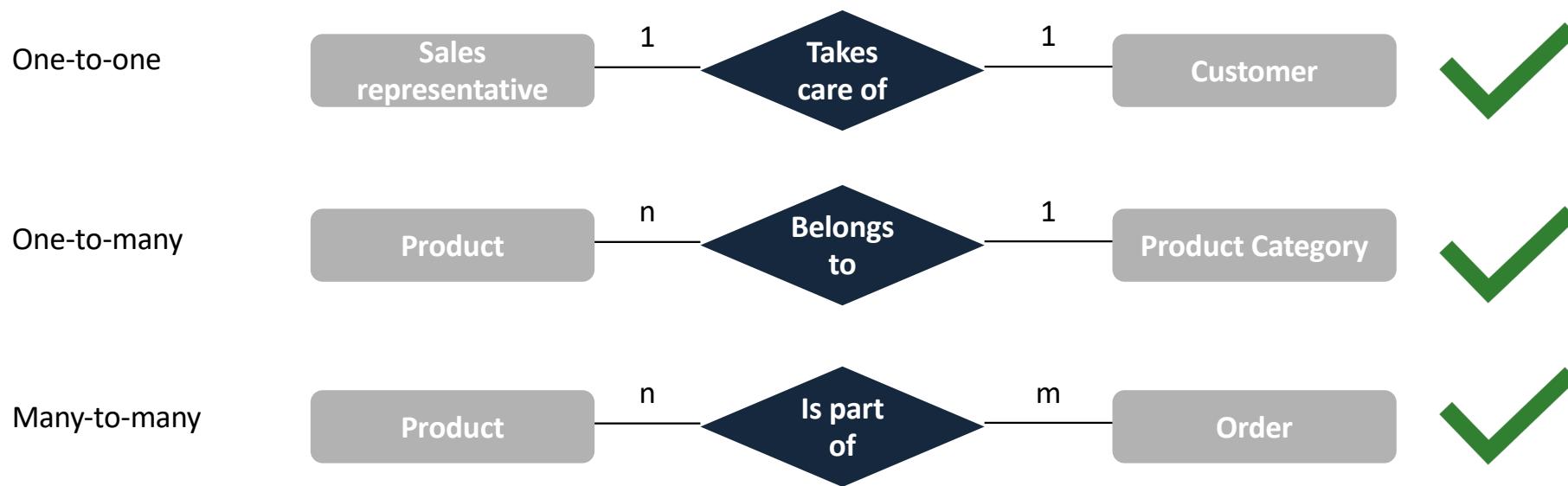
Network Database Model Emerged From Hierarchical Database Model



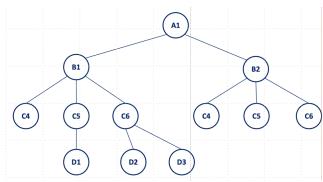
Network Database Model Emerged From Hierarchical Database Model



Network Data Model Can Handle All Three Relations



Conceptual Data Models



Hierarchical Data Model



Network Data Model

order_id	product_id	customer_id	amount	date	...
251	2	55	100	2024-02-12	...
id	name	weight	color
1	shorts	20	green
2	shoe	550	black
...

Relational Data Model

Relational Database Model

Order Table

order_id	product_id	customer_id	amount	date	...
251	2	55	100	2024-02-12	...

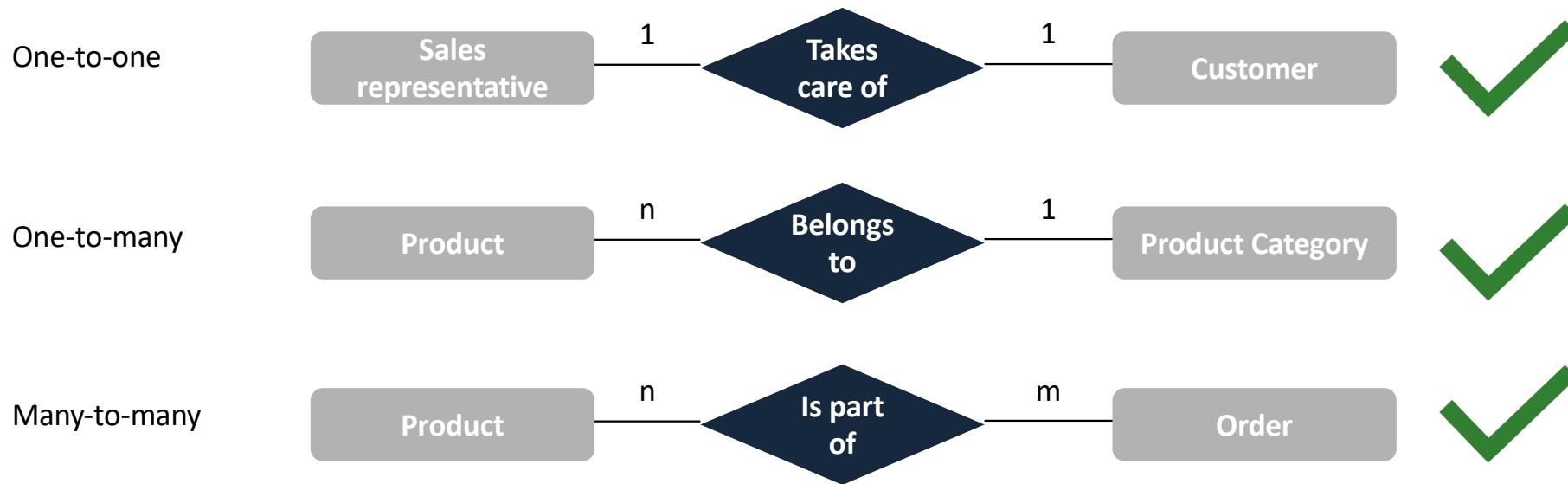
Product Table

id	name	weight	color	...
1	shorts	20	green	...
2	shoe	550	black	...
...

Customer Table

id	name	city	Email	...
1	Wang	Beijing	wang@...	...
...
55	Smith	New York	smith@...	...

Many-to-many Relationships Can Not Be Handled Well with Hierarchical Data Models



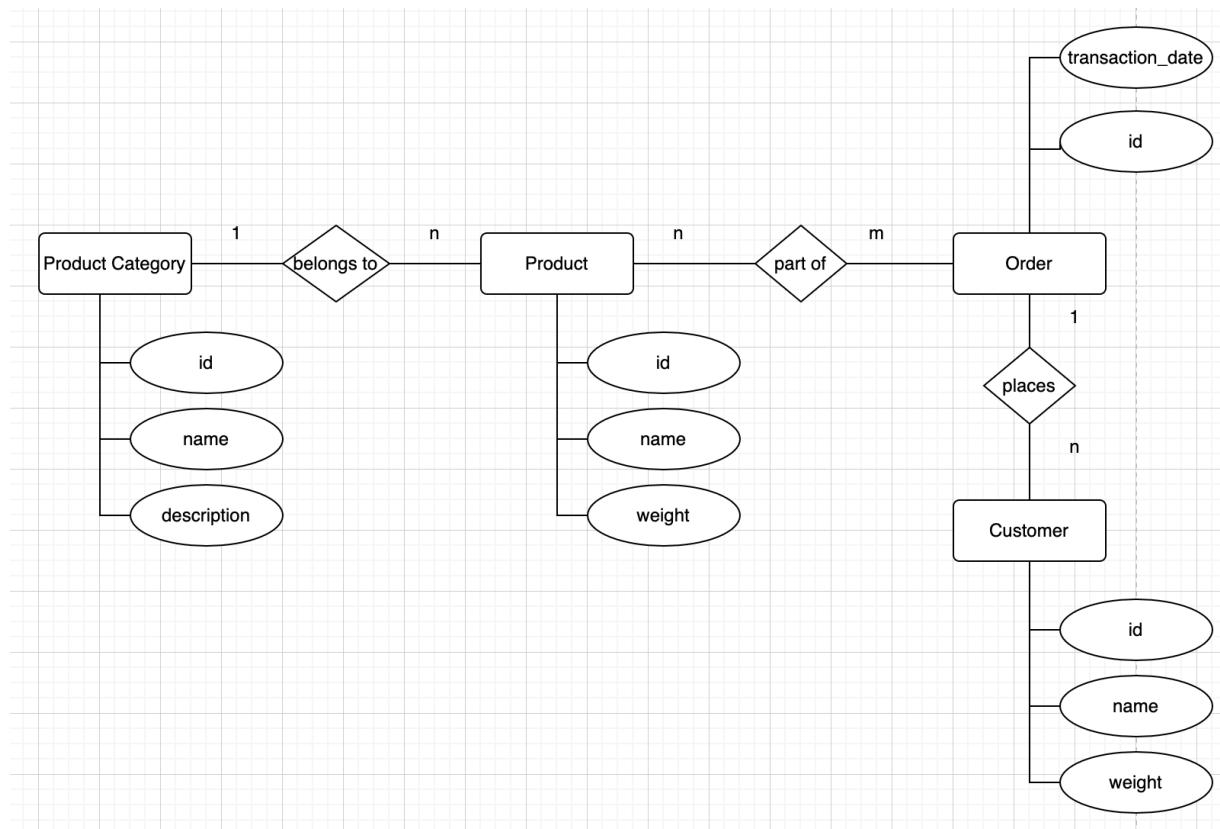
Agenda

Business Analytics

- Introduction
- Types of Data and Databases
- Data Models
- Entity Relationship Model**
- Case 1: Reverse Engineer a Database Model
- The Star and Snowflake Schema
- Analyzing Data with Power BI
- Case 2: Analyzing Procurement Transactions

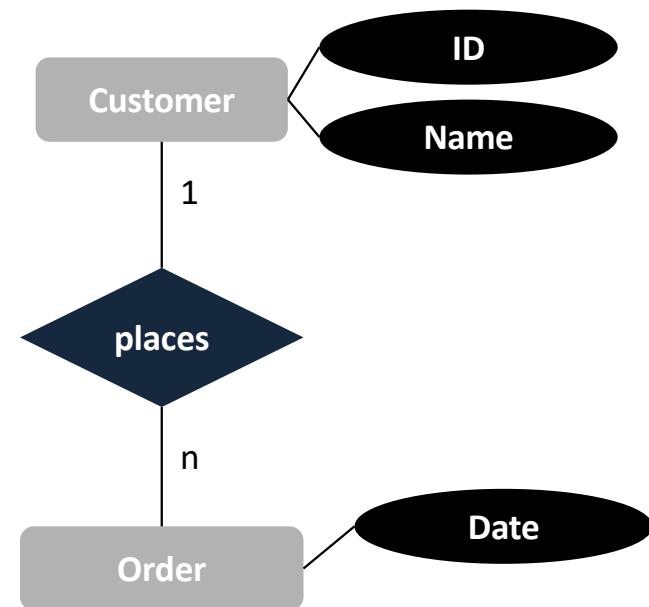
Data Science

The Entity-relationship Approach Is an Example for a Conceptual Data Model

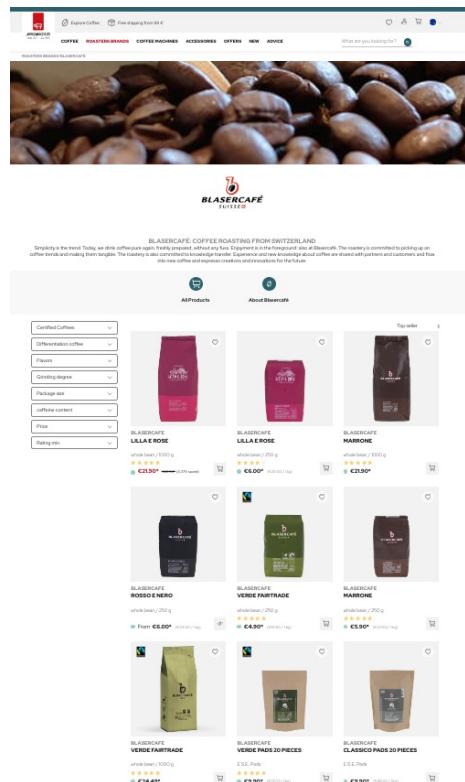


The Entity-relationship Approach Is an Example for a Conceptual Data Model

- E/R diagrams allow to visualize an organization's data elements as well as the relationships between them
- **Rectangles** identify **entities** about which the organization collects data
- **Diamonds** enclose and name **relationships** between entities
- **Lines** connecting entities to the diamond show whether the relationship is exclusive or not (cardinality)
- **Ellipses** show **attributes** of entities and relationships



An Entity Is a Real-world Class of Things Such as Product or Customer

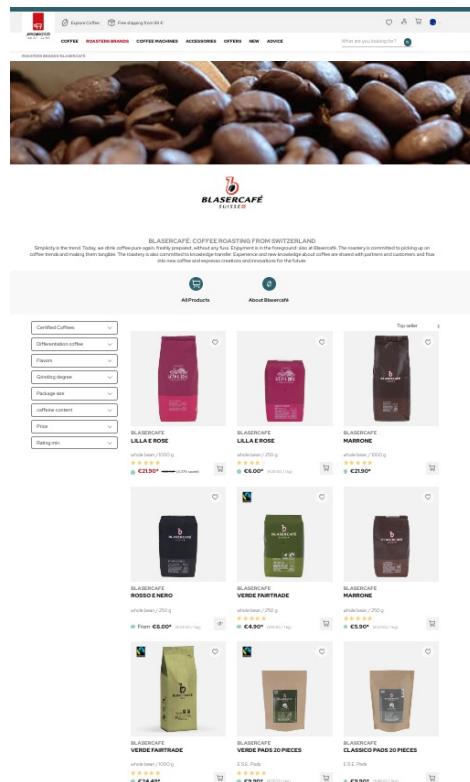


Product

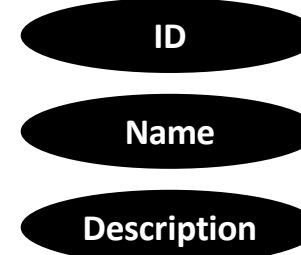
Customer

Order

What Are the Entities in This Diagram?



Product



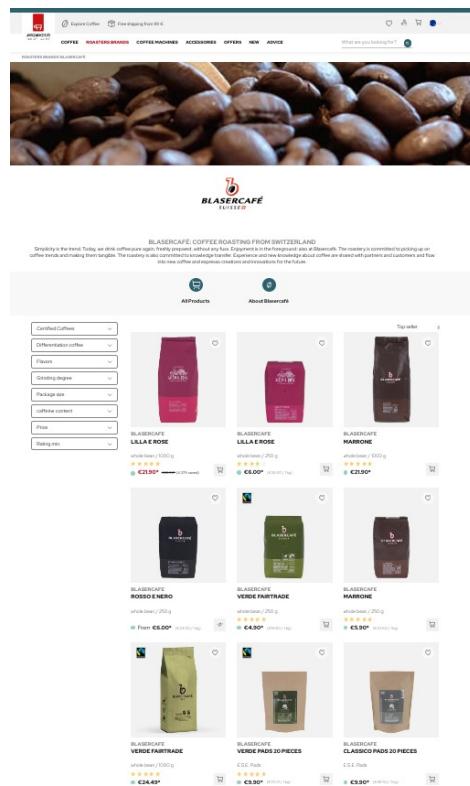
ID	Name	Description
...

Customer

Description

Order

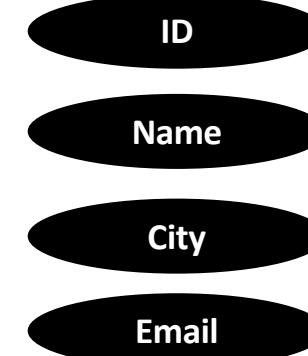
What Are the Entities in This Diagram?



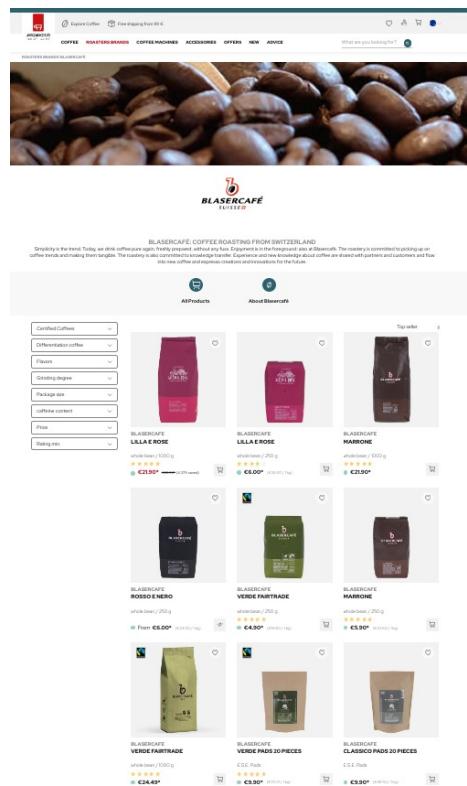
Product

Customer

Order



Attributes Typically Correspond to Columns



Product

Customer

Order

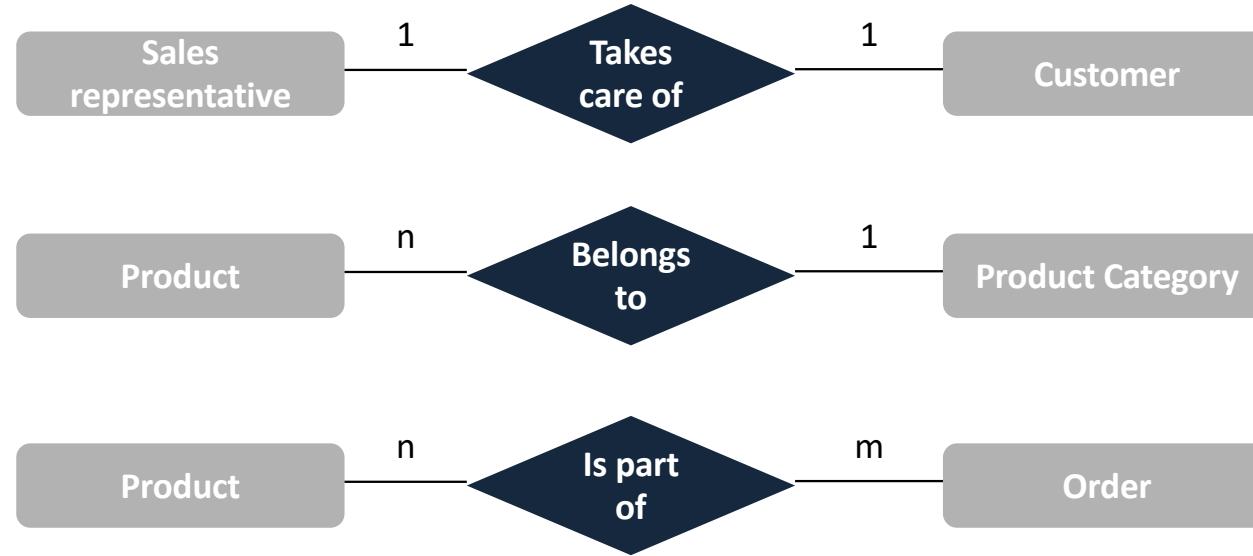
ID

Date

Amount

ID	Date	Amount
...

Types of Relations - Cardinalities



Types of Relations - Cardinalities



ID	Last Name	...
1	Manson	...
2	Kim	...
3	Johnson	...

ID	Last Name	Sales Rep.	...
1	Muram	3	...
2	Jameson	2	...
...

Types of Relations - Cardinalities



ID	Name	Category	...
1	Harmony	2	...
2	New York	1	...
3	Jamaican	3	...
...

ID	Name	...
1	Standard	...
2	Premium	...
3	Luxury	...
...

Types of Relations – Cardinalities N:m



ID	Name	Category	...
1	Harmony	2	...
2	New York	1	...
3	Jamaican	3	...
...

ID	Date	Amount	Products
1	2023-11-01	150	[1, 3]
2	2023-06-15	200	[4, 6]
3	2023-08-31	250	1
...

Types of Relations – Cardinalities N:m

Product Table

ID	Name	Category	...
1	Harmony	2	...
2	New York	1	...
3	Jamaican	3	...
...

Order Table

ID	Date	Amount
1	2023-11-01	150
2	2023-06-15	200
3	2023-08-31	250
...

Types of Relations – Cardinalities N:m



Product Table

ID	Name	Category	...
1	Harmony	2	...
2	New York	1	...
3	Jamaican	3	...
...

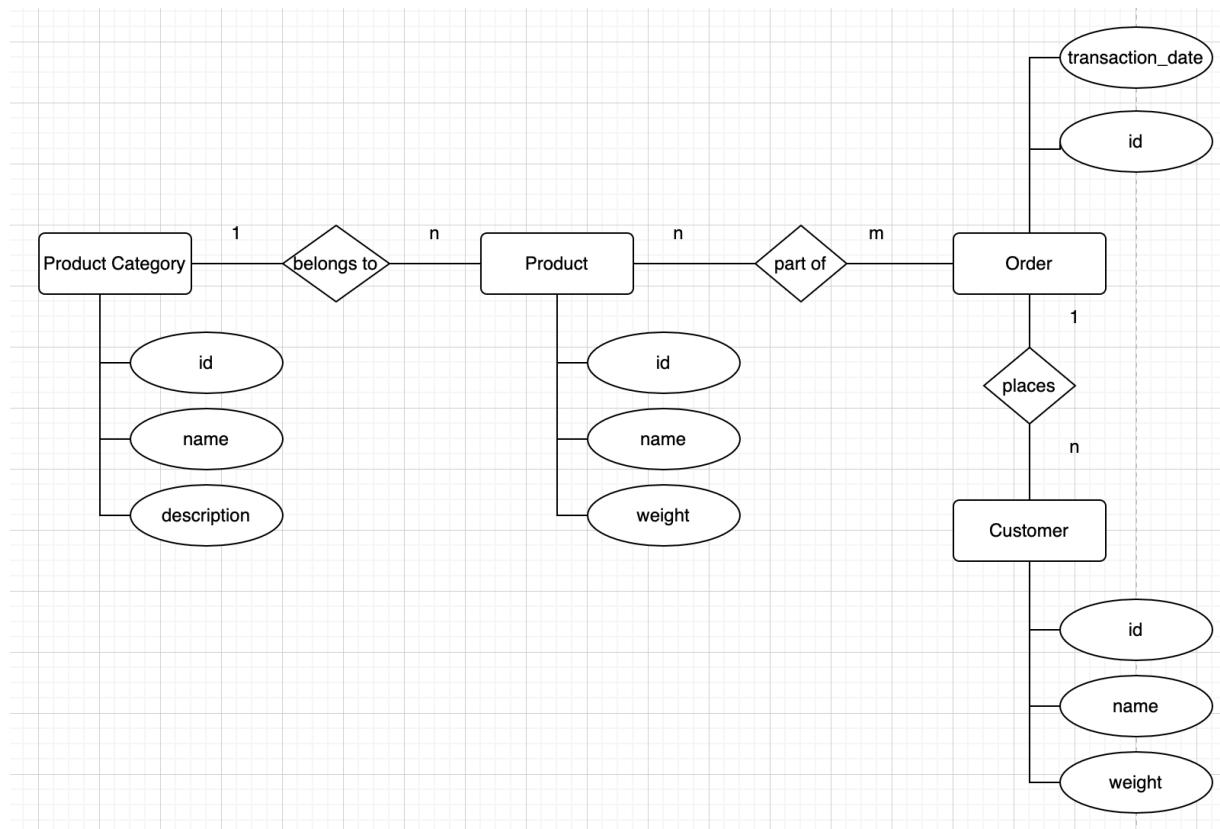
Order Lines Table

ID	OrderID	ProductID	Kg
1	1	1	10
2	1	2	20
3	1	5	2
...

Order Table

ID	Date	Amount
1	2023-11-01	150
2	2023-06-15	200
3	2023-08-31	250
...

There Is One Error in This Diagram Š



Agenda

Business Analytics

Introduction

Types of Data and Databases

Data Models

Entity Relationship Model

Case 1: Reverse Engineer a Database Model

The Star and Snowflake Schema

Analyzing Data with Power BI

Case 2: Analyzing Procurement Transactions

Data Science

Case 1: Reverse Engineer a Database Model

You are provided with five xlsx / csv files:

- customers.xlsx / customers.csv
- products.xlsx / customers.csv
- order_items.xlsx / order_items.csv
- sales_orders.xlsx / orders.csv
- products.xlsx / products.csv

Your task is to build an Entity-Relationship Model:

1. Identify the entities which need to be included
2. Identify the primary and foreign keys which can be used to link the tables
3. Identify the type of cardinality between the entities
4. Add the attributes to each entity

A Comment on Real World Products Keys



Key	value
GTIN*	4005900933034
DM-Article	710736
Amazon-Article	B09XF4T5V8

*Global Trade Item Number

Generating Unique Keys with Universally Unique Identifier (uuid)

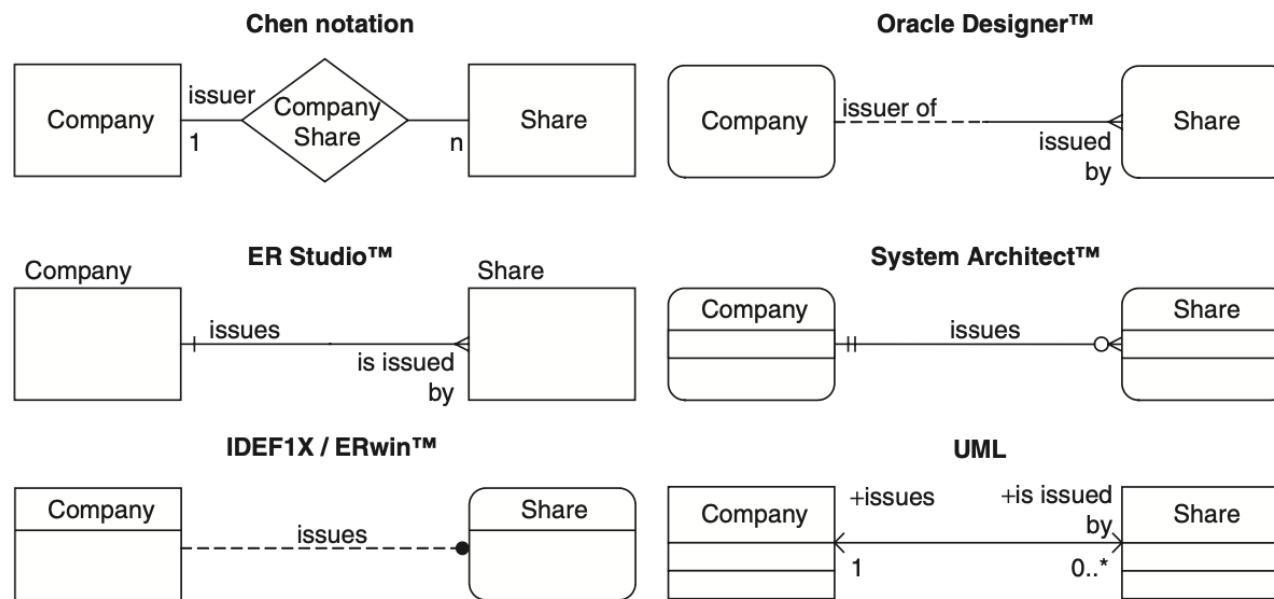
Example UUID: 13806bf6-1541-479c-94ed-ba0fd2bb05f5

Online-Generator: <https://www.uuidgenerator.net/>

Python:

```
import uuid
uuid.uuid4()
```

Examples for Other Types of Notations



The Logical Data Model Also Defines the Types of Data in Each Table

Product table

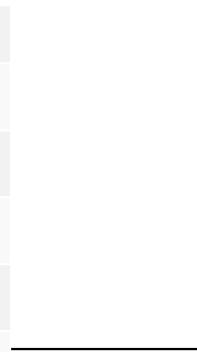
name	type
id 	integer
name	text
description	text
weight	float
price	float
categoryl_id	integer

Category table

name	type
id 	integer
name	text
description	text

Primary Key

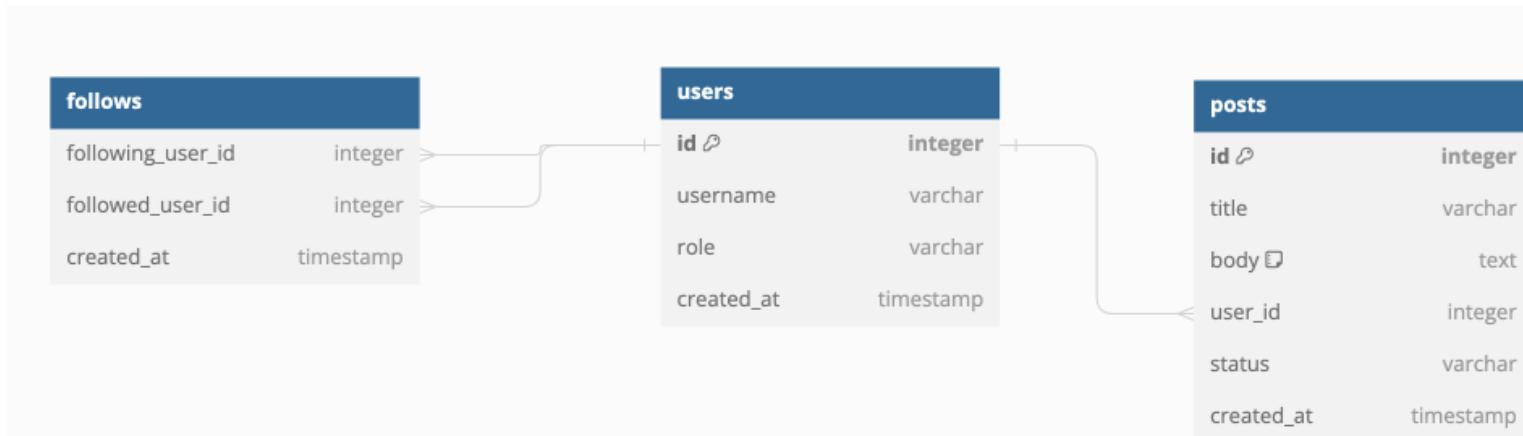
Foreign Key



Case 1: Reverse Engineer a Database Model

Please highlight the primary and secondary key in your database model

Tools – for the Building a Logical Data Model - Dbdiagram



<https://dbdiagram.io>

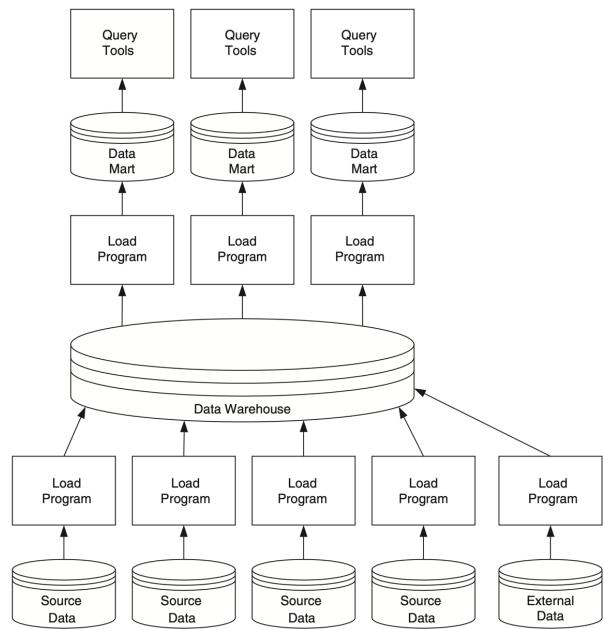
Agenda

Business Analytics

- Introduction
- Types of Data and Databases
- Data Models
- Entity Relationship Model
- Case 1: Reverse Engineer a Database Model
- The Star and Snowflake Schema
- Analyzing Data with Power BI
- Case 2: Analyzing Procurement Transactions

Data Science

Data Warehouses and Data Marts



Key Features of data marts

- Present data in a form that is understandable for people in business functions
- Often structured by business domain, e.g., sales, procurement, logistics, finance
- Common data structure for Tableau, PowerBI or Qlik reporting (dashboards)
- Two common data modeling schemas:
 - **star schema**
 - **snowflake schema**

Simsion & Witt, p. 474

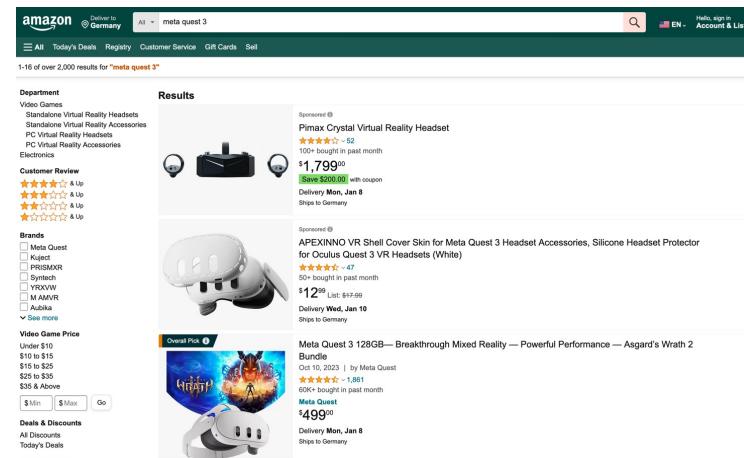
Star and a Snowflake Schemas Consist of Two Types of Tables:

- Fact Tables
- Dimension Tables

A Fact Tables Consist of Measurements, Metrics or Facts of a Business Process, E.g., Sales Transactions

Fact Tables

Dimension Tables



#	Date	Amount	Product	Customer	...
06B5278A-...	2024-02-23	499.99	B0C8VKH1ZH	5CD779AB-...
...

A Fact Tables Consist of Measurements, Metrics or Facts of a Business Process, E.g., Sales Transactions

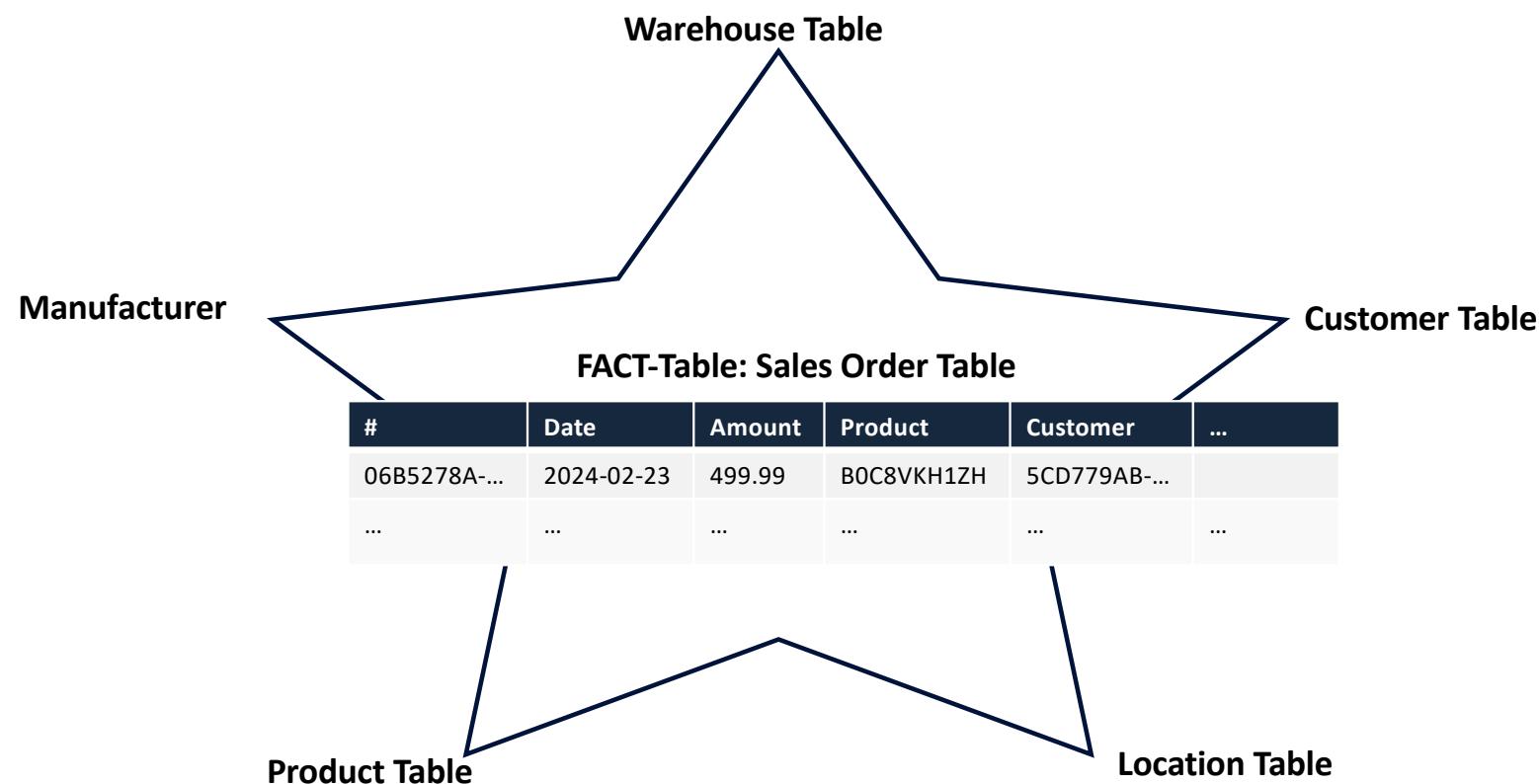
■ Fact Tables

■ Dimension Tables

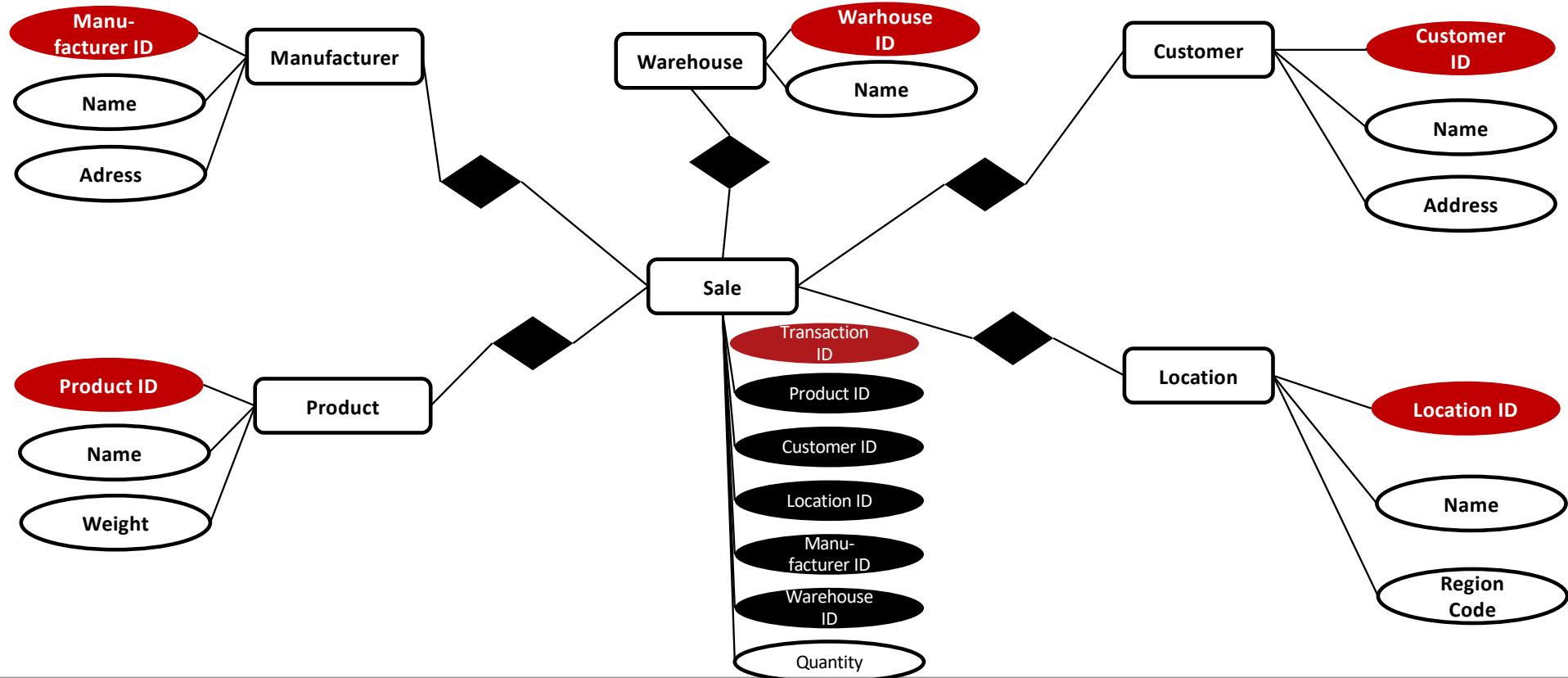
Manufacturer ID	Name	City	Postal Code	Contact	...
OCF1EA9C-...	Meta		

Customer ID	Name	City	Postal Code	Contact	...
7Xq1DP13-...	Wang		

Star-schema



Star-schema



Summary Star-schema

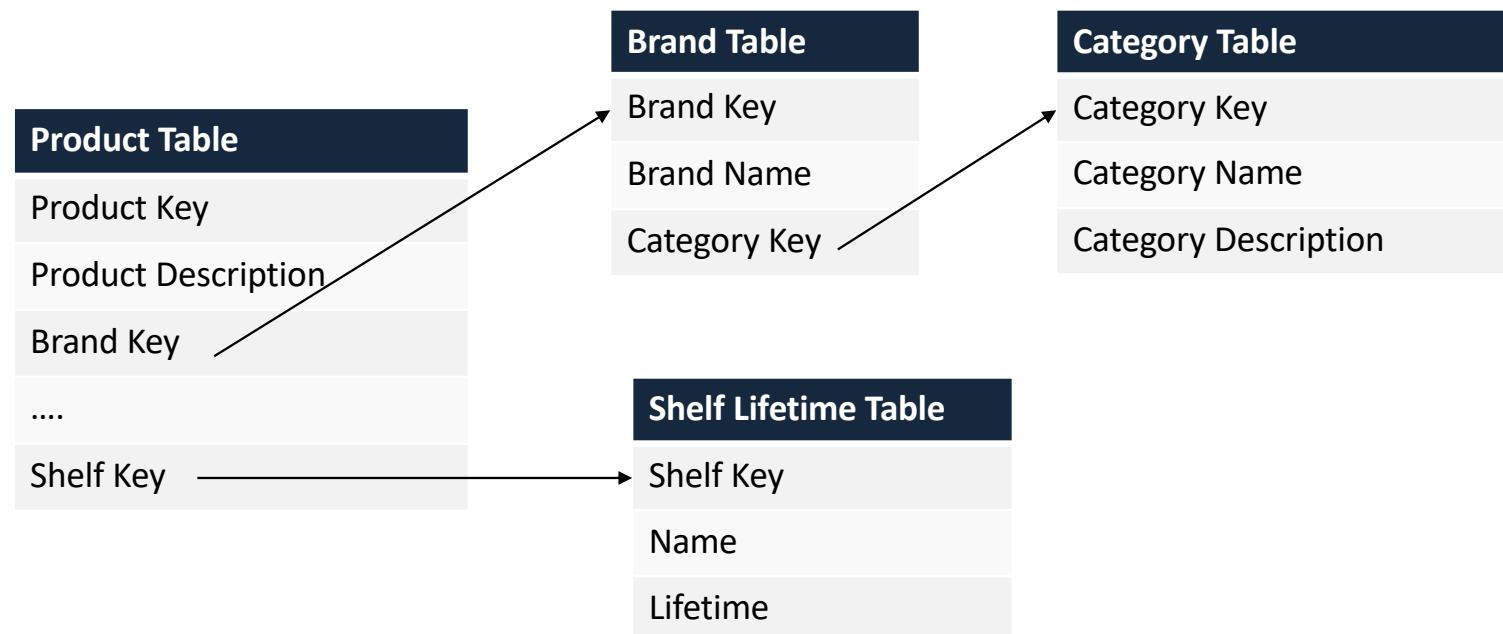
- Common data modeling approach for Data-Marts, organize data in a way that it can be understood by people from the business functions
- Common approach for data models behind dashboard (e.g., PowerBI, Tableau, Qlik)
- Fact table is surrounded by several dimensions table
- Only one fact table can be part of the star schema
- True star schema only supports one level of dimension

Product Table in a Star-schema

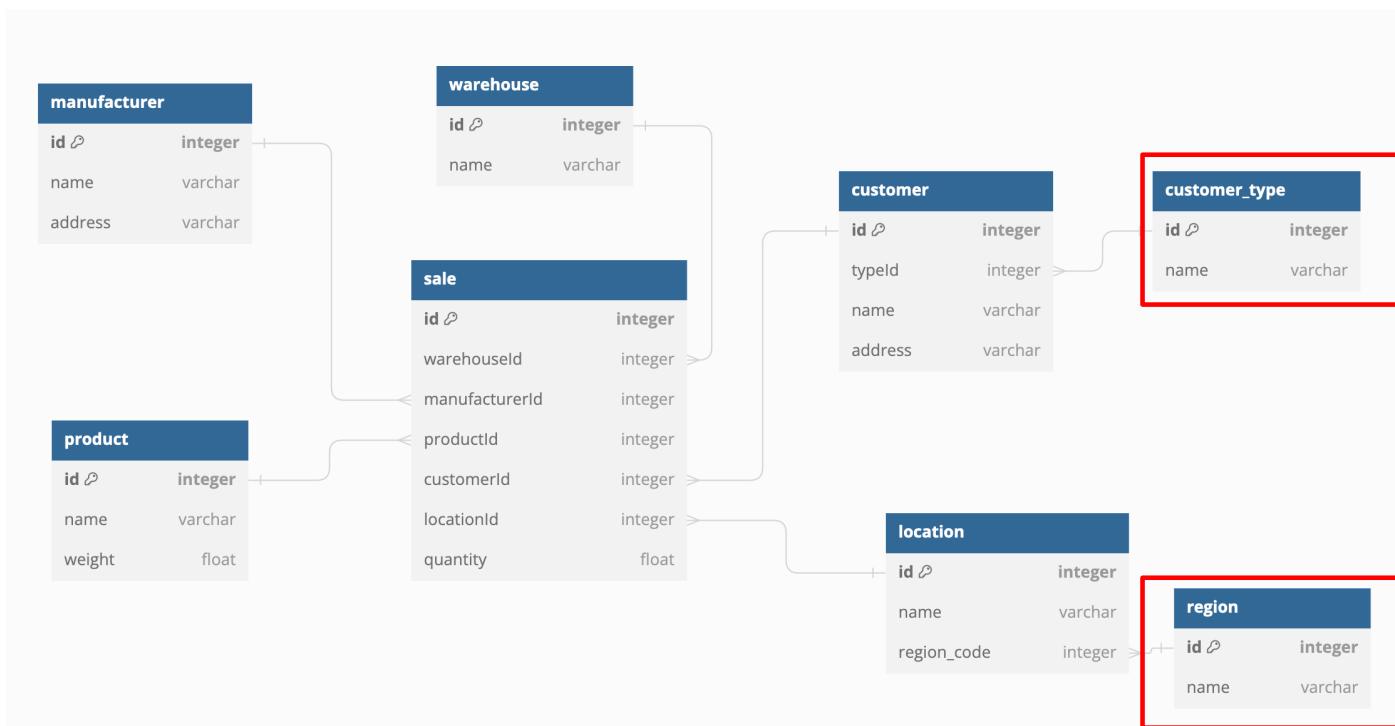


Columns
Product Key
Product Description
Category Name
Brand Name
Storage Type
Package Type
Shelf Lifetime
Weight

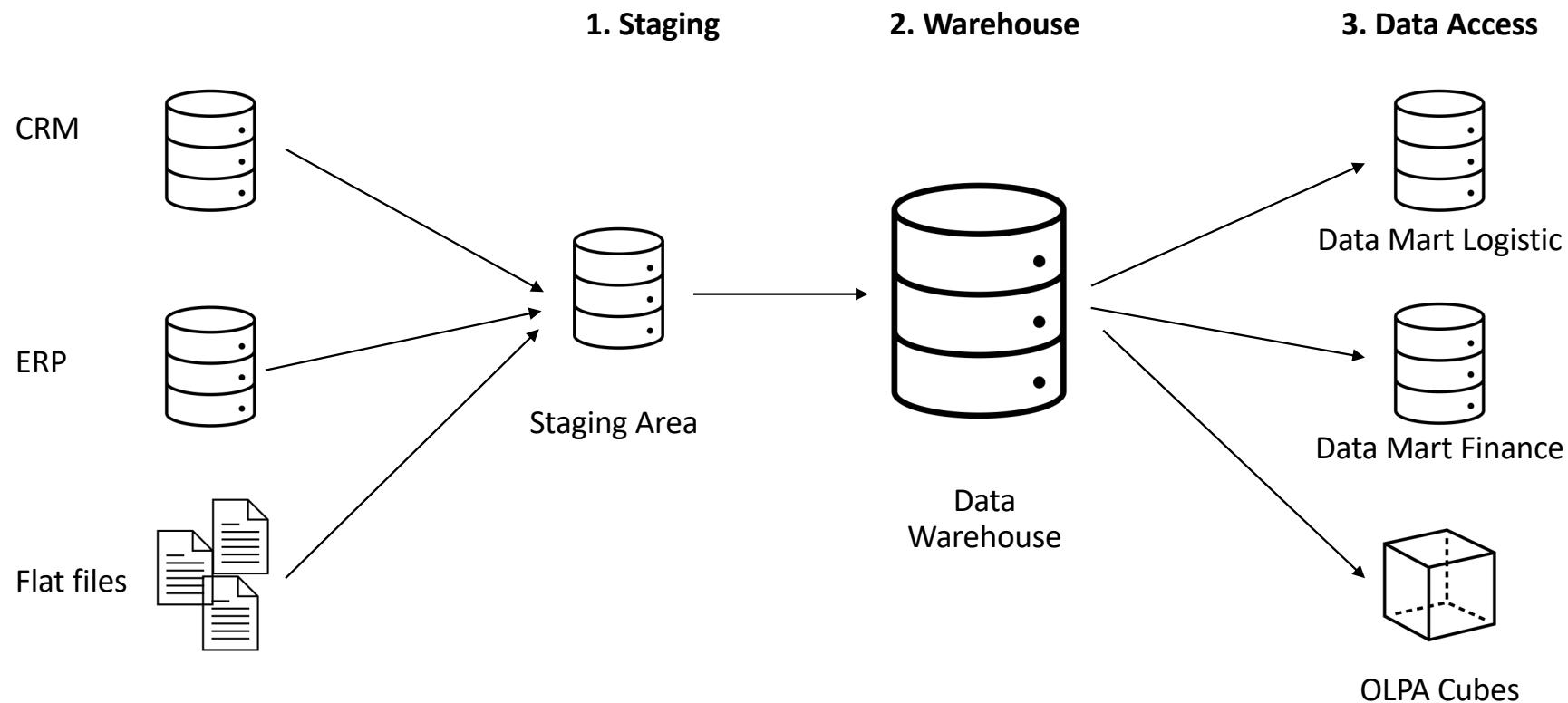
Snowflaked Product Dimensions



Snowflaked Product Dimensions



Typical 3-layer Architecture



Source: Abramson, I. Data Warehouse: The Choice of Inmon vs. Kimball. IAS Inc.

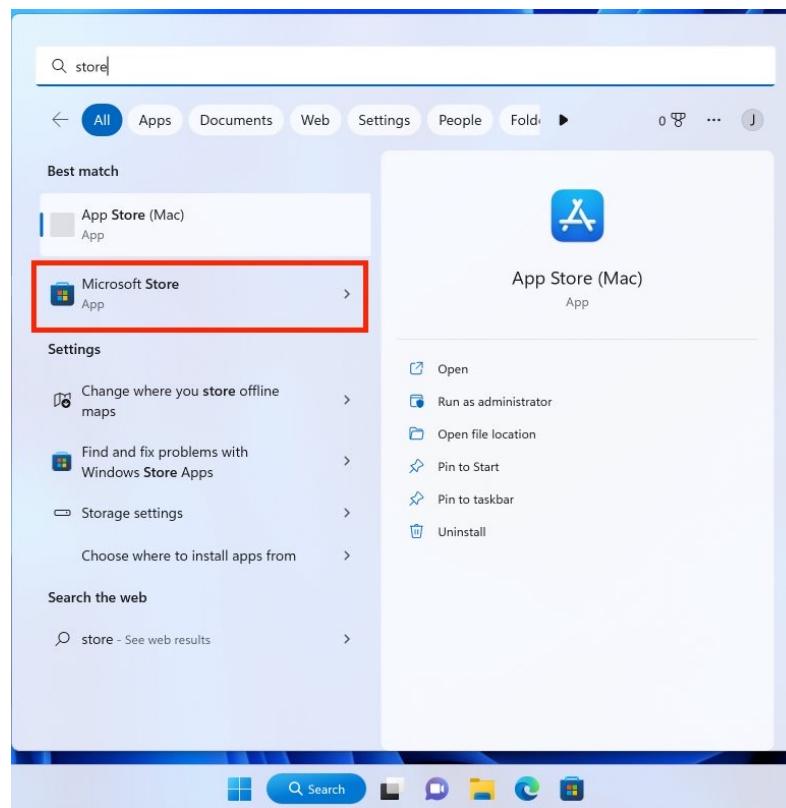
Agenda

Business Analytics

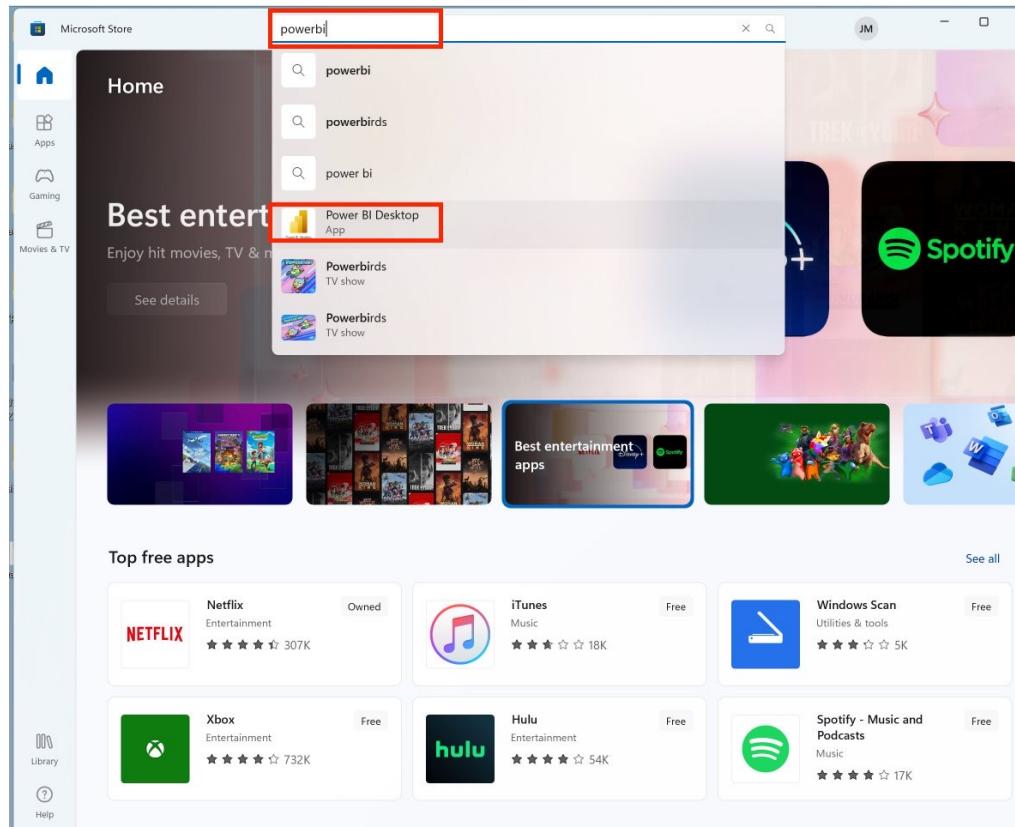
- Introduction
- Types of Data and Databases
- Data Models
- Entity Relationship Model
- Case 1: Reverse Engineer a Database Model
- The Star and Snowflake Schema
- Analyzing Data with Power BI
- Case 2: Analyzing Procurement Transactions

Data Science

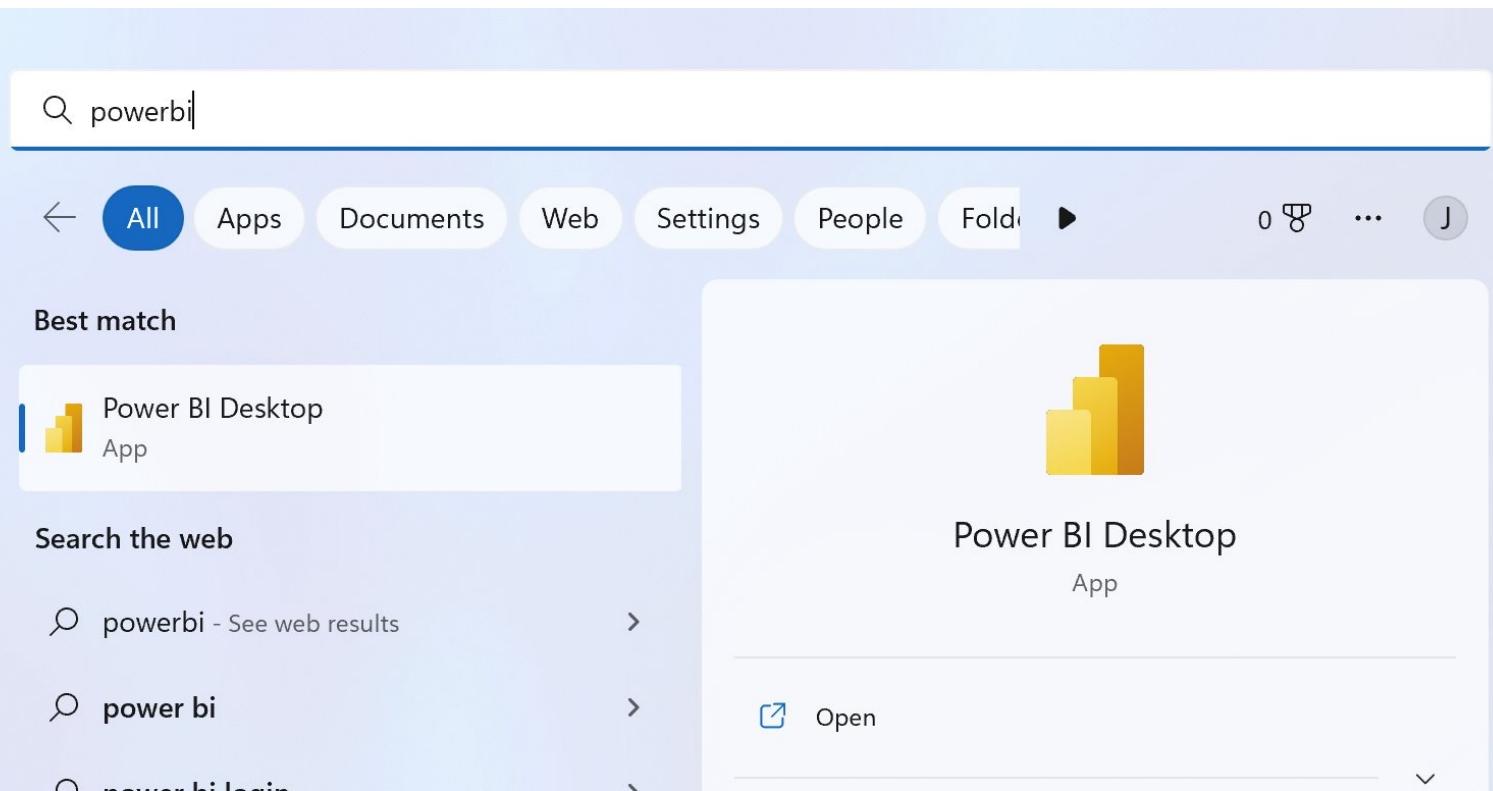
Installation (1): Open Microsoft Store



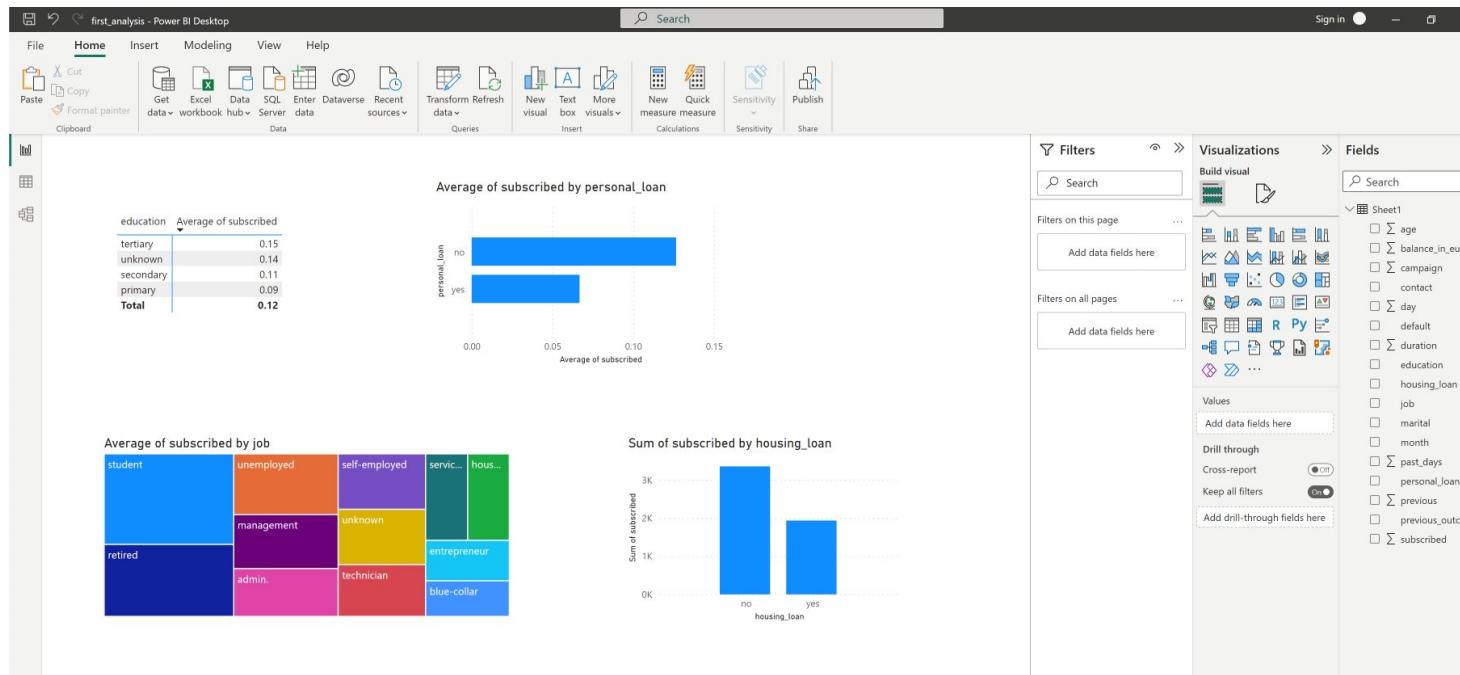
Installation (2): Search for “powerbi Desktop_”



Installation (3): Verify Installation

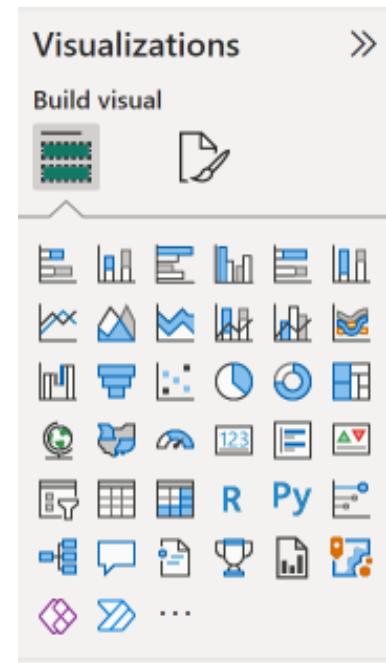


Creating Visuals



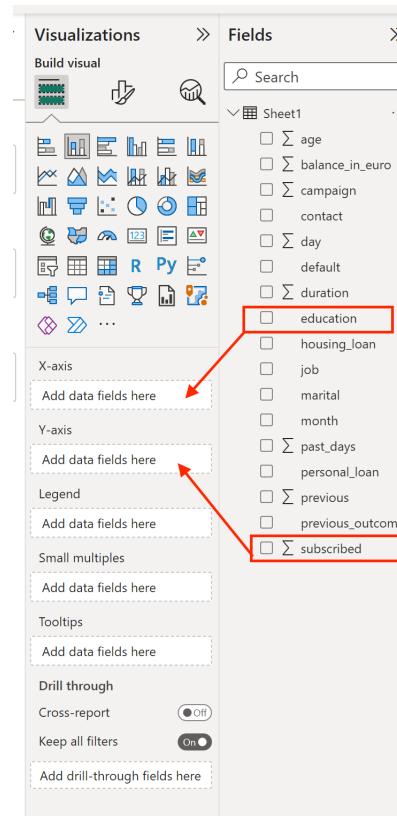
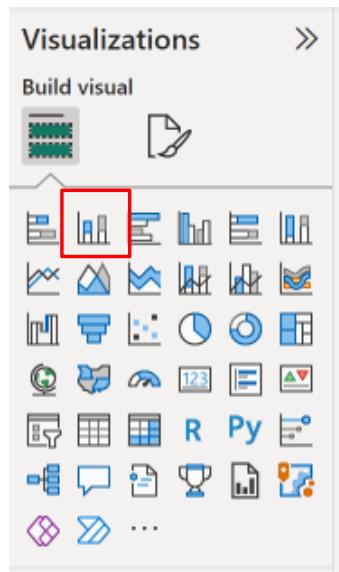
Creating Visuals

- Power BI has several types of visuals integrated
- The market place offers further visuals



Creating Visuals – Stacked Column Chart

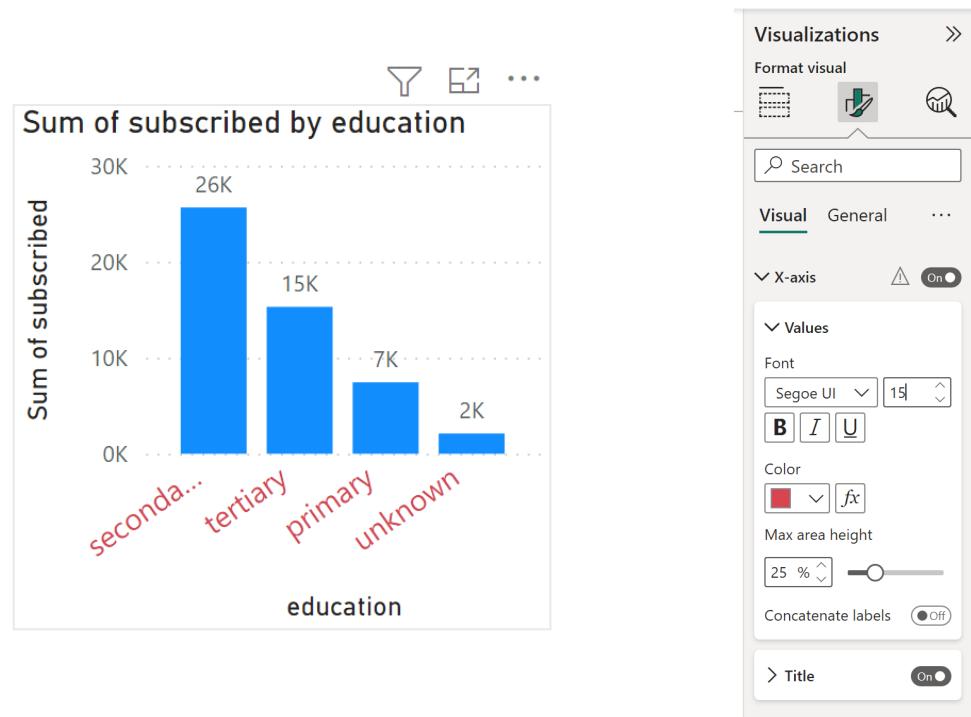
1 Select stacked column chart



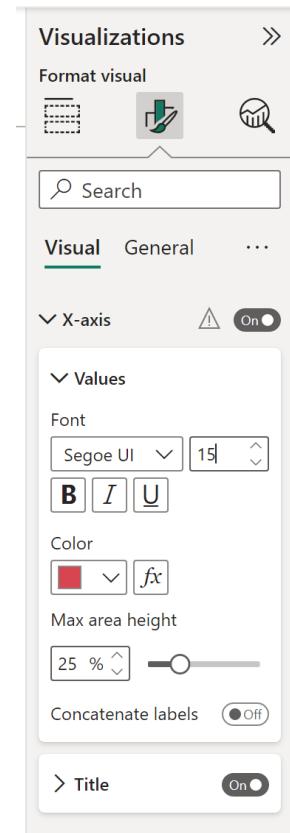
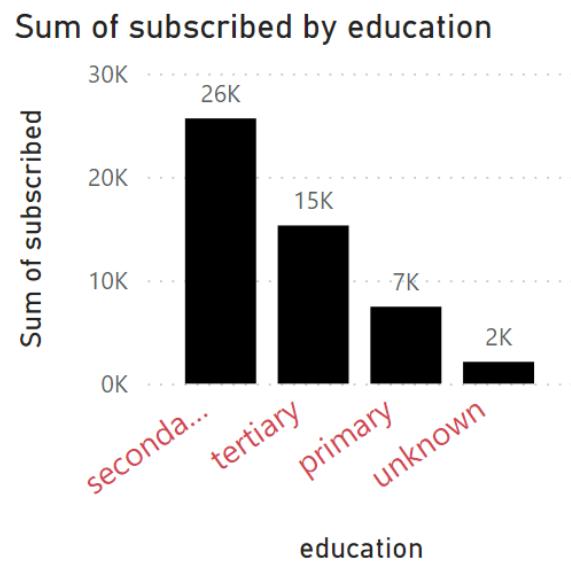
2 Drag & Drop education on x-axis

3 Drag & Drop subscribed on y-axis

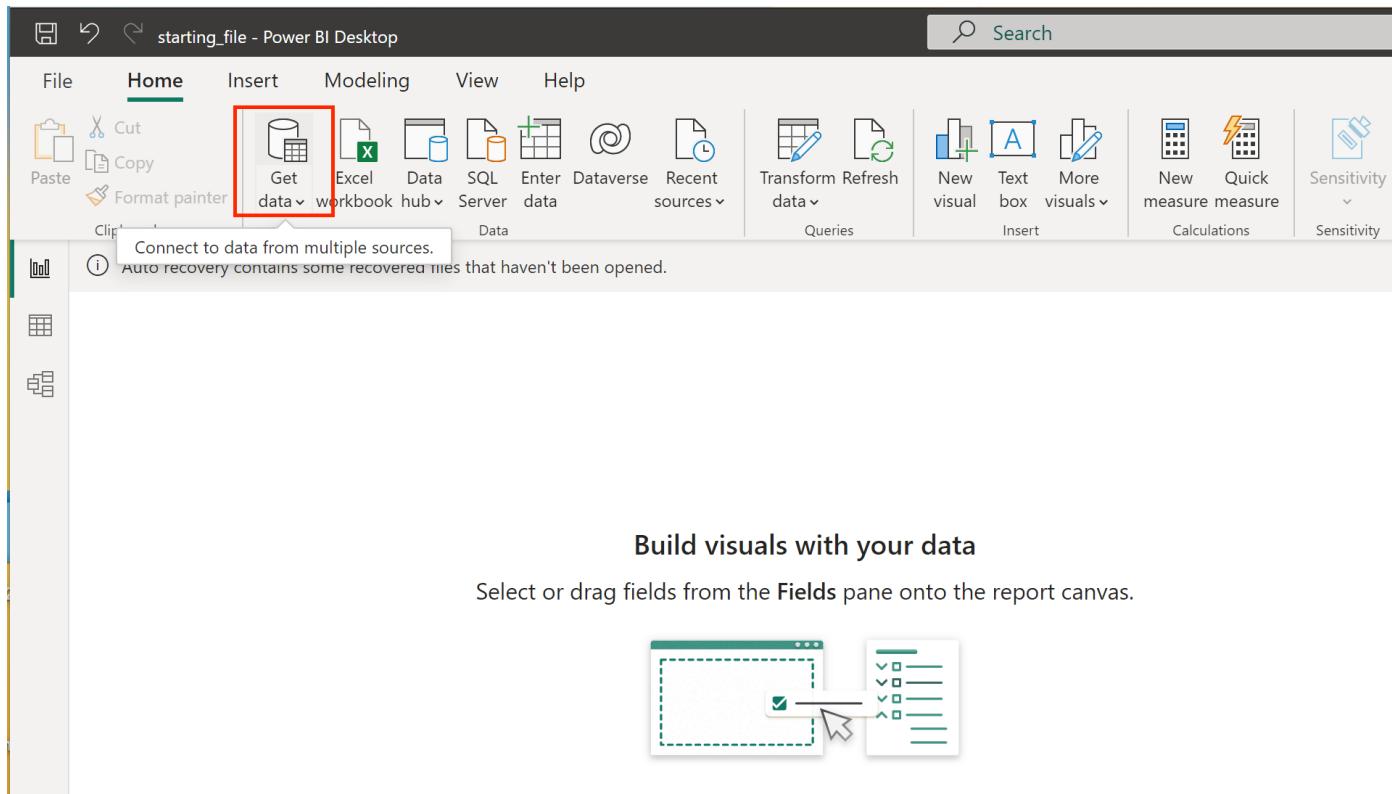
Creating Visuals – Stacked Column Chart



Creating Visuals – Stacked Column Chart



Loading Data



Loading Data – Ms Excel File

The screenshot shows the Microsoft Power BI Data Load Preview window. On the left, the 'Navigator' pane displays a folder structure with 'bank_marketing.xlsx [1]' and 'Sheet1' selected. The main area, titled 'Sheet1', shows a table with the following columns: age, job, marital, education, default, and balance_in_euro. The data consists of 45 rows of customer information. At the bottom right of the preview window are three buttons: 'Load' (green), 'Transform Data' (grey), and 'Cancel' (grey).

age	job	marital	education	default	balance_in_euro
58	management	married	tertiary	no	
44	technician	single	secondary	no	
33	entrepreneur	married	secondary	no	
47	blue-collar	married	unknown	no	
33	unknown	single	unknown	no	
35	management	married	tertiary	no	
28	management	single	tertiary	no	
42	entrepreneur	divorced	tertiary	yes	
58	retired	married	primary	no	
43	technician	single	secondary	no	
41	admin.	divorced	secondary	no	
29	admin.	single	secondary	no	
53	technician	married	secondary	no	
58	technician	married	unknown	no	
57	services	married	secondary	no	
51	retired	married	primary	no	
45	admin.	single	unknown	no	
57	blue-collar	married	primary	no	
60	retired	married	primary	no	
33	services	married	secondary	no	
28	blue-collar	married	secondary	no	
56	management	married	tertiary	no	
32	blue-collar	single	primary	no	

Loading Data – Ms Excel File

The screenshot shows the Microsoft Power Query Editor interface. The ribbon at the top has tabs for File, Home, Transform, Add Column, View, Tools, and Help. The Home tab is selected. The main workspace shows a table with the following data:

	age	job	marital	education	default
1		management	married	tertiary	no
2	58	technician	single	secondary	no
3	44	entrepreneur	married	secondary	no
4	33	blue-collar	married	unknown	no
5	47	unknown	single	unknown	no
6	33	management	married	tertiary	no
7	35	management	single	tertiary	no
8	28	entrepreneur	divorced	tertiary	yes
9	42	retired	married	primary	no
10	58	technician	single	secondary	no
11	43	admin.	divorced	secondary	no
12	41	admin.	single	secondary	no

The 'Transform' tab is selected in the ribbon, and the 'Transform' pane on the right shows the following steps:

- Source
- Navigation
- Promoted Headers
- Changed Type

A 'Query Settings' pane on the right shows the sheet name as 'Sheet1'.

Loading Data – Ms Excel File

The screenshot illustrates the process of loading data from an Excel file. On the left, a 'Datatype' dialog box is open, showing various options for the 'age' column: Decimal Number, Fixed decimal number, Whole Number, Percentage, Date/Time, Date, Time, Date/Time/Timezone, Duration, Text, True/False, Binary, and Using Locale... The 'Text' option is selected. In the center, a small preview of the data shows three rows with values 1, 2, and 3 in the first column and 58, 44, and 33 in the second column. The 'age' header is highlighted with a red box. On the right, a 'Filter Data' dialog box is open, listing numbers from 22 to 39, each preceded by a checked checkbox. The 'OK' button is visible at the bottom right of the dialog.

Datatype

1 2 3	age	A B C
1.2	Decimal Number	
\$	Fixed decimal number	
1 2 3	Whole Number	
%	Percentage	
⌚	Date/Time	
📅	Date	
🕒	Time	
⌚📍	Date/Time/Timezone	
⌚	Duration	
A B C	Text	
✗✓	True/False	
☰	Binary	
Using Locale...		

Filter Data

- Sort Ascending
- Sort Descending
- Clear Sort
- Clear Filter
- Remove Empty
- Number Filters

Search:

- (Select All)
- 22
- 23
- 24
- 25
- 26
- 27
- 28
- 29
- 30
- 31
- 32
- 33
- 34
- 35
- 36
- 37
- 39

⚠ List may be incomplete. Load more

OK Cancel

Agenda

Business Analytics

- Introduction
- Types of Data and Databases
- Data Models
- Entity Relationship Model
- Case 1: Reverse Engineer a Database Model
- The Star and Snowflake Schema
- Analyzing Data with Power BI

Data Science

Case 2: Build a BI-report in Power Bi

You are tasked with building a Business Intelligence report for the Procurement Department of Rustic Roasters, a coffee trading company. The company is experiencing significant challenges with late purchase orders arriving at their warehouses. The Procurement Department wants to gain deeper insights into this issue and identify potential causes for these delays.

You have been provided with the following data files:

- customer.xlsx
- material.xlsx
- logistics_partners.json
- procurement_transactions_part_A.xlsx
- procurement_transactions_part_B.xlsx
- supplier.xlsx
- Territory.xlsx
- warehouse.xlsx

Your report should leverage these datasets to analyze the current situation and support data-driven decision making for the Procurement Department.

Case 2: Build a BI-report in Power Bi – Task Part 2

- Many companies faced significant transport challenges during the COVID-19 pandemic. Did Rustic Roasters also experience similar issues? Please analyze the data to identify any pandemic-related disruptions.
- Additionally, there was a workers' strike in the past. Can you determine which warehouse(s) were affected by the strike?

Agenda

Business Analytics

- Data Science**
 - Classification vs. Regression
 - Prediction Models
 - Case 3: Building a Prediction Model for Purchase Order Delivery
 - Measuring the Quality of a Prediction Model
 - Feature Engineering
 - Explainable AI

Agenda

Business Analytics

Data Science

Classification vs. Regression

Prediction Models

Case 3: Building a Prediction Model for Purchase Order Delivery

Measuring the Quality of a Prediction Model

Feature Engineering

Explainable AI

Predictive Analytics - Target

Past:

Order	Confirmed Date	Delivered Date
#1	2024-06-14	2024-06-18
#2	2024-07-16	2024-07-25
#3	2024-07-14	2024-07-14

Present:

Order	Confirmed Date	Delivered Date
#1	2025-08-14	?
#2	2025-09-16	?
#3	2025-09-14	?

Models

-  Linear and Logit Regression
-  Decision Tree
-  Random Forest
-  Boosted Trees
-  Neural Network

Classification Vs Regression

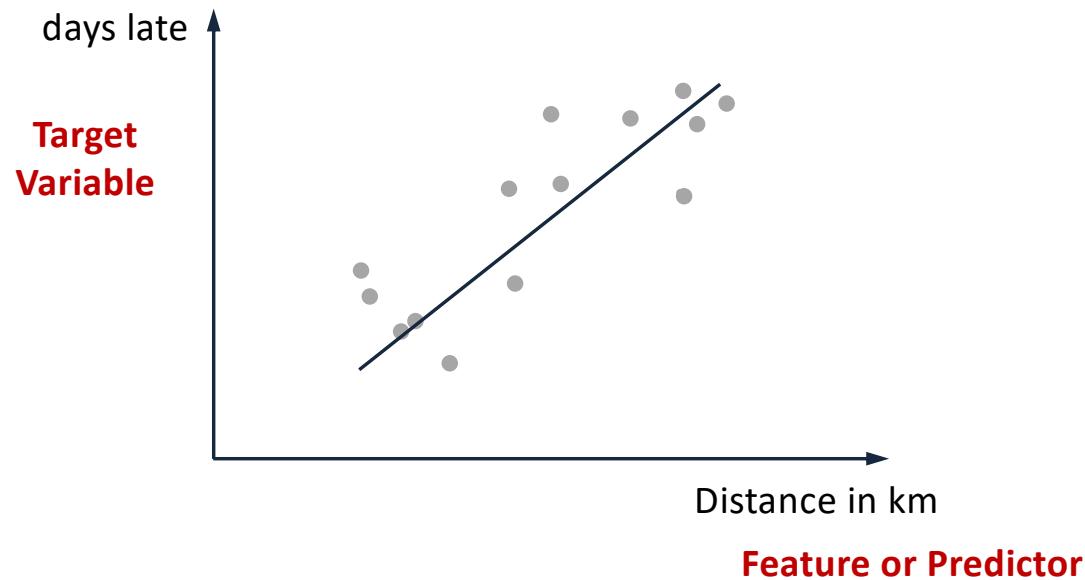
Present:

Order	Confirmed Date	Delivered Date	Late	Days late
#1	2025-06-14	?	?	?
#2	2025-07-16	?	?	?
#3	2025-07-14	?	?	?

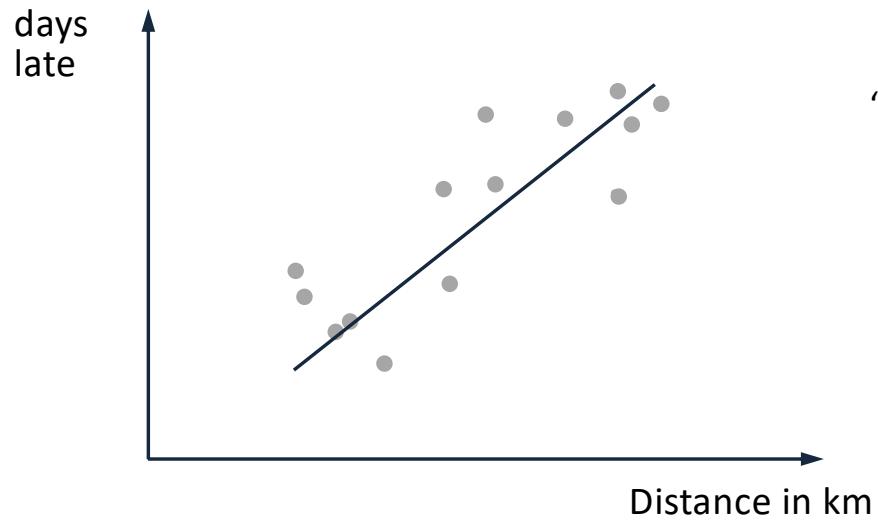
Which One Is a Classification Problem?

- Predict raw material consumption to optimize stocks
- Predict risk of suppliers going bankrupt
- Predict if a picture shows a t-shirt

Regression



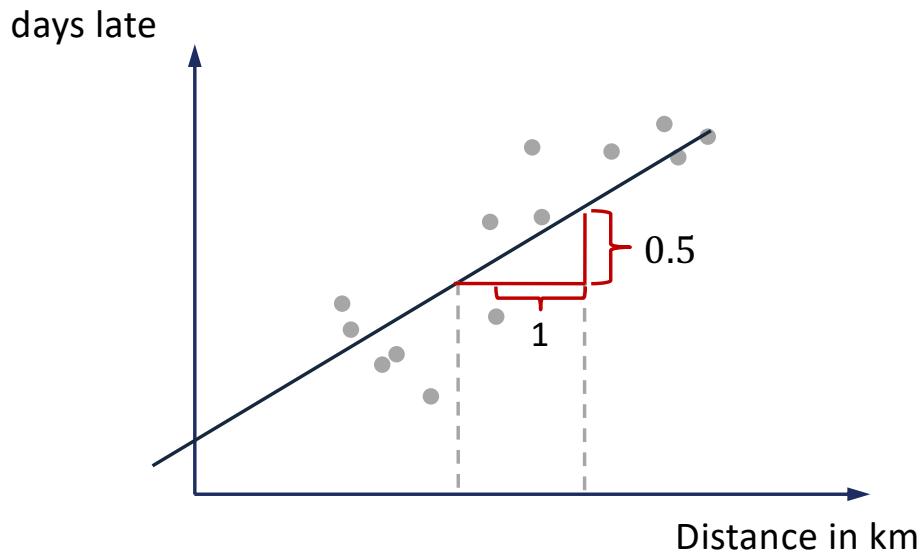
Example Regression Analysis



"Abstract Model": $y = \beta_0 + \beta_1 x$

Specific Model: $days\ late = \beta_0 + \beta_1 Distance\ in\ km$

Example Regression Analysis

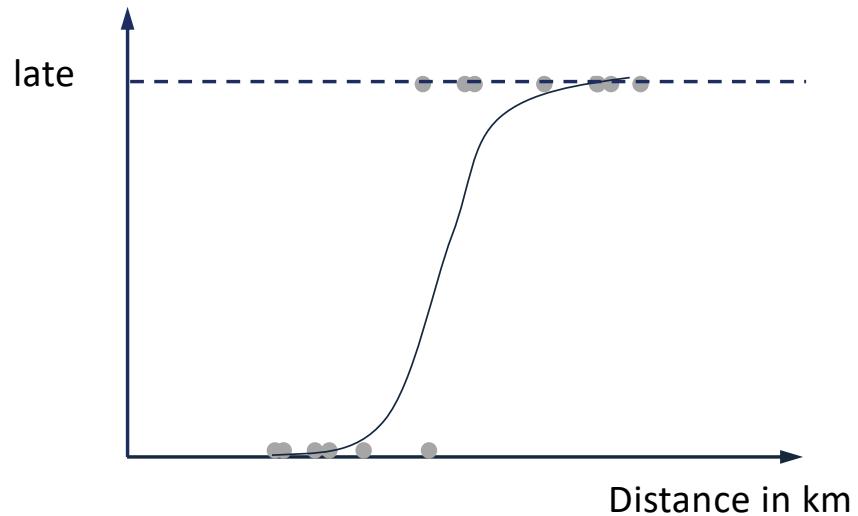


Specific Model with estimated parameters:

$$\text{days late} = 0.1 + 0.5 \text{ Distance in km}$$

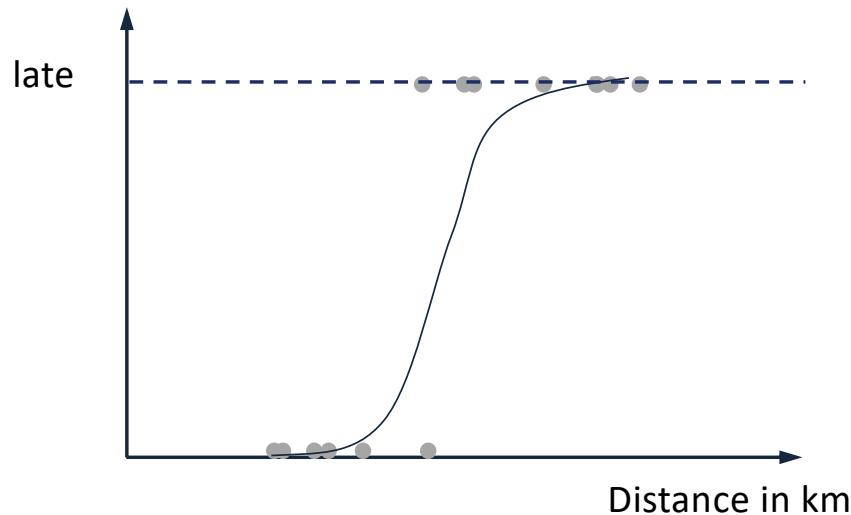
Order ID	Distance in km	Predicted days late
1	10	5.1
2	5	2.6
...

Classification



$$p(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

Classification



$$p(x) = \frac{e^{0.1+0.5\text{Distance in km}}}{1+e^{0.1+0.5\text{Distance in km}}}$$

Order ID	Distance in km	Predicted probability late
1	10	84%
2	5	72%
...

Agenda

Business Analytics

Data Science

Classification vs. Regression

Prediction Models

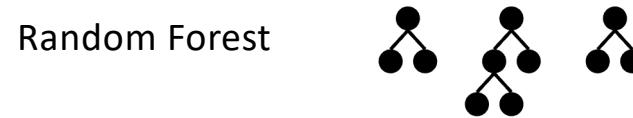
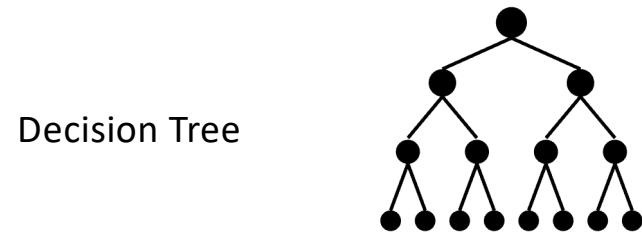
Case 3: Building a Prediction Model for Purchase Order Delivery

Measuring the Quality of a Prediction Model

Feature Engineering

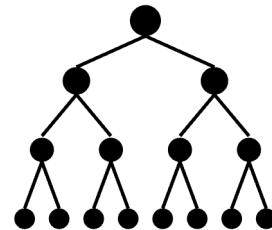
Explainable AI

Ensemble Learning



Ensemble Learning

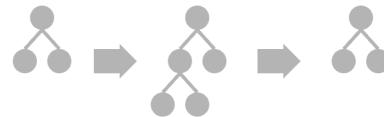
Decision Tree



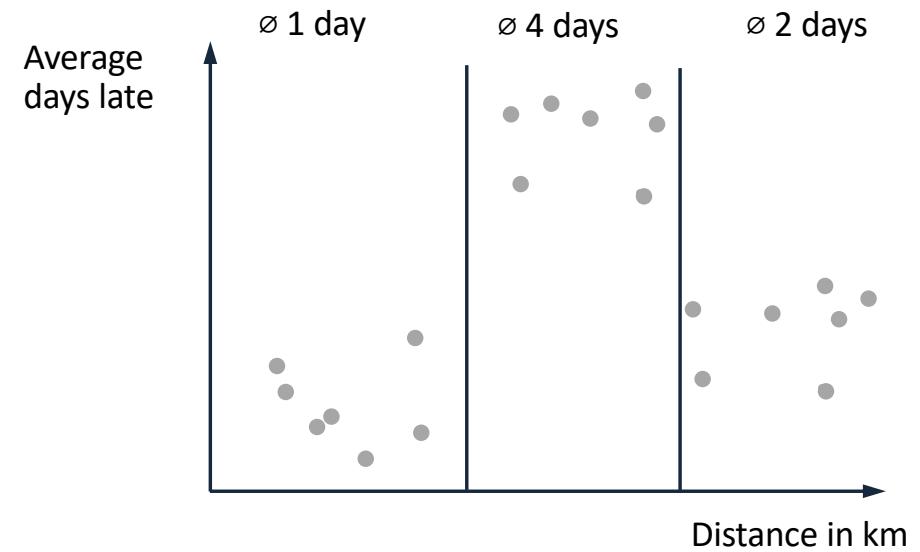
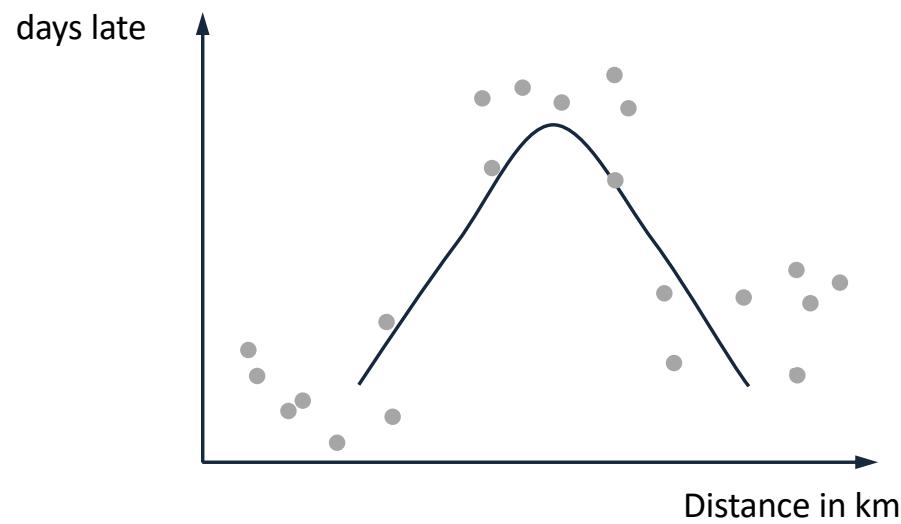
Random Forest



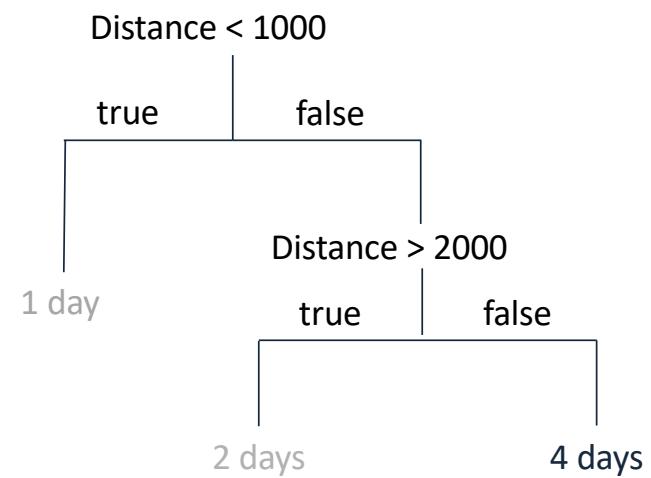
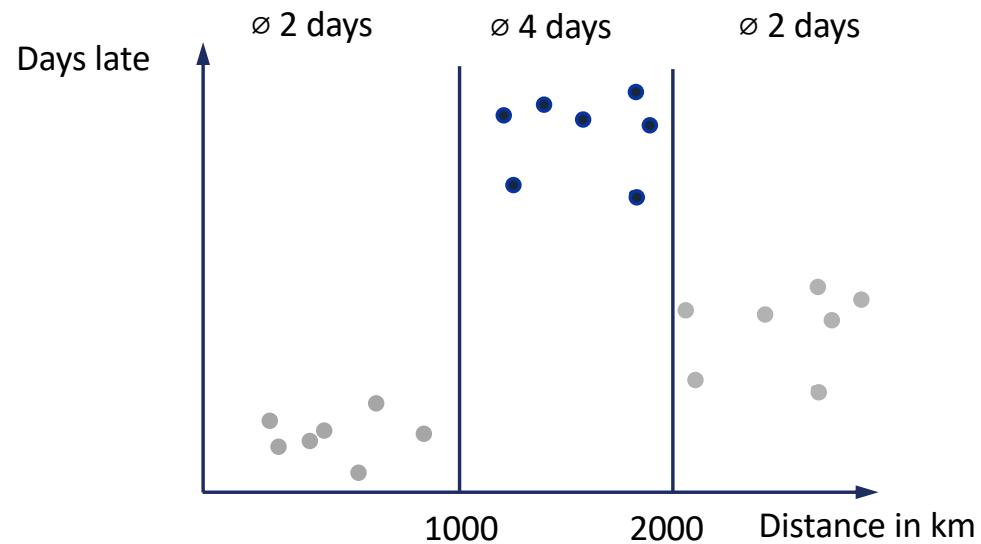
Boosted Trees



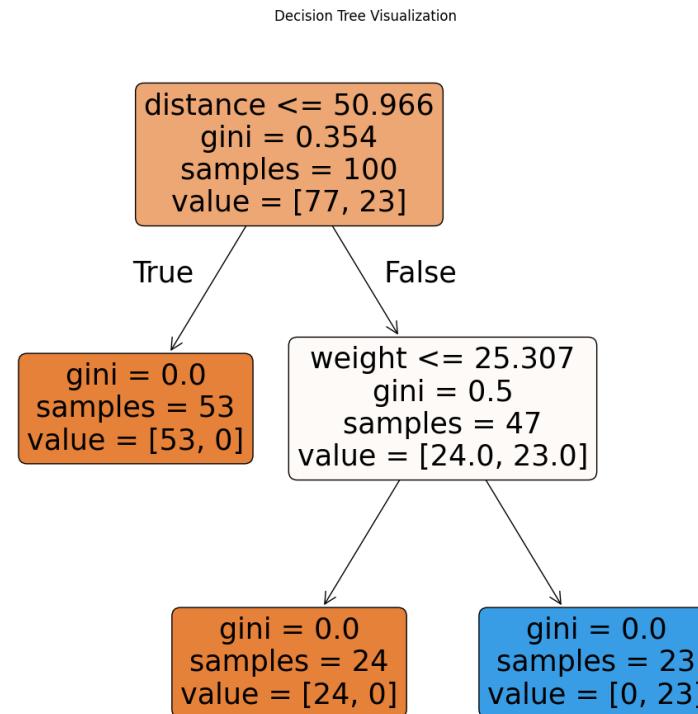
Limitations of Linear and Logistic Regression



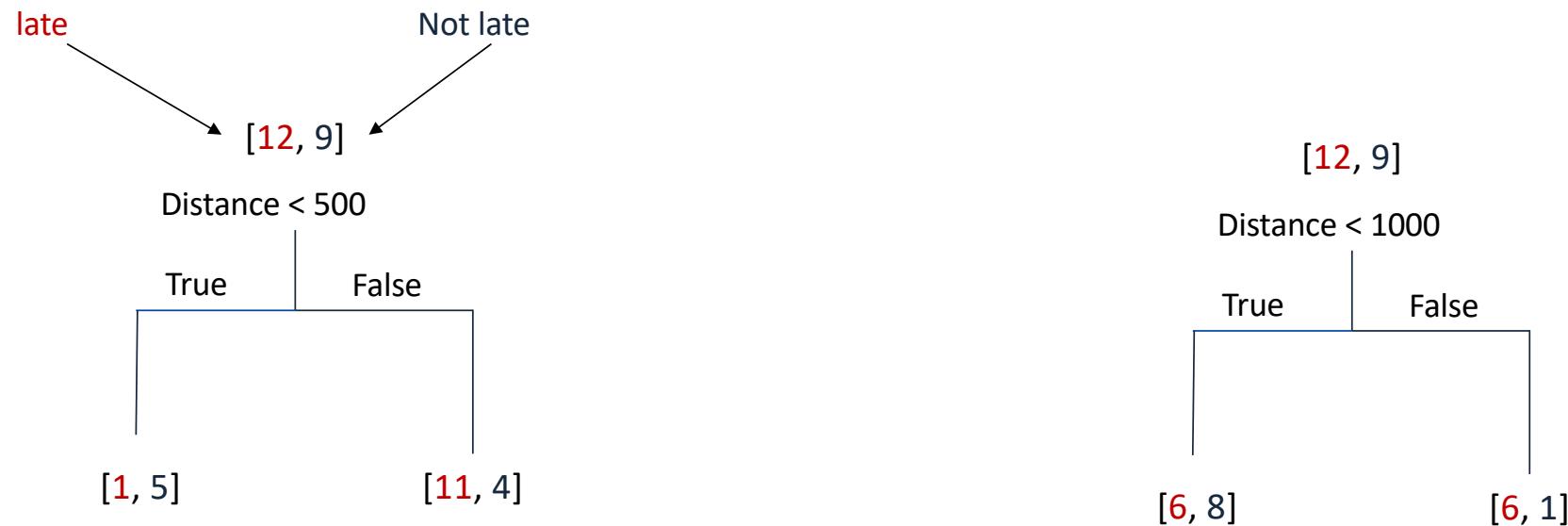
Decision Tree for Regression



Example for Decision Tree in Python



Decision Tree for Classification - Finding Splits



Does the split improve our classification?

Which split is better?

Decision Tree for Classification - Finding Splits

$$Gini \ index = \sum_{k=1}^K \hat{p}_k(1 - \hat{p}_k)$$

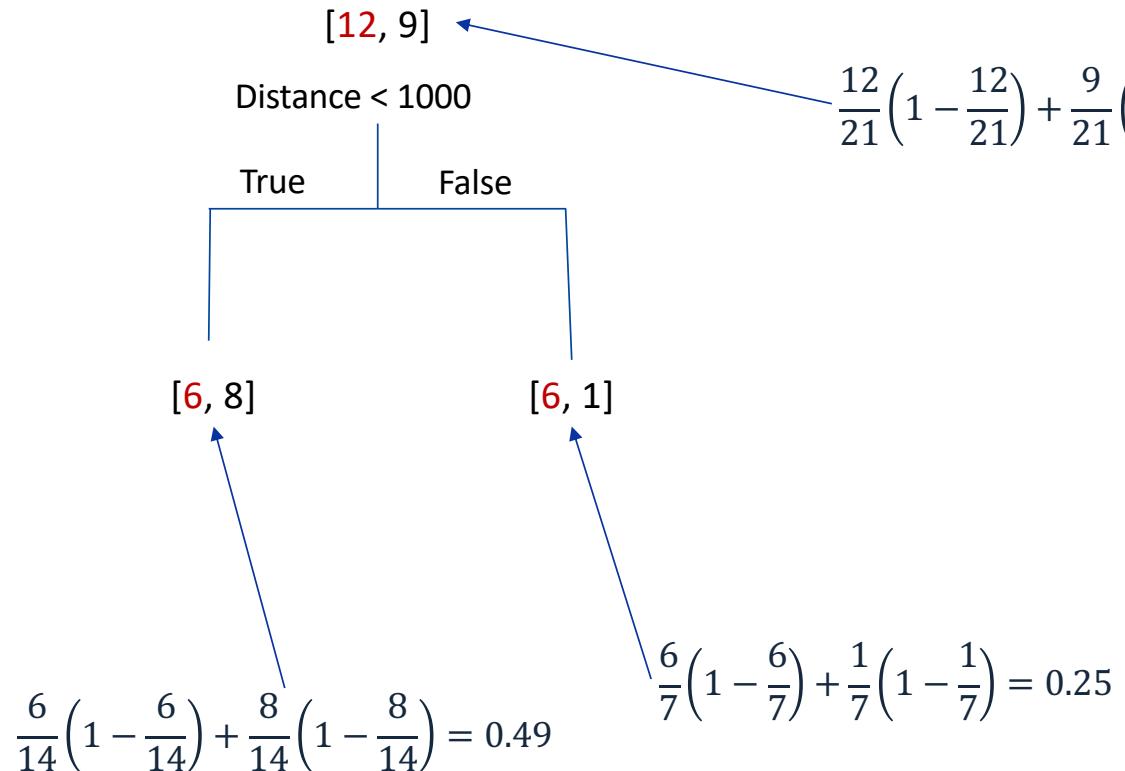
[True(s), False(s)]

$$[4, 0] \quad Gini \ index = \frac{4}{4}\left(1 - \frac{4}{4}\right) + \frac{0}{4}\left(1 - \frac{0}{4}\right) = 0$$

$$[2, 2] \quad Gini \ index = \frac{2}{4}\left(1 - \frac{2}{4}\right) + \frac{2}{4}\left(1 - \frac{2}{4}\right) = \frac{1}{2}$$

$$[1, 3] \quad Gini \ index = \frac{1}{4}\left(1 - \frac{1}{4}\right) + \frac{3}{4}\left(1 - \frac{3}{4}\right) = \frac{3}{8}$$

Decision Tree for Classification - Finding Splits



$$\frac{12}{21} \left(1 - \frac{12}{21}\right) + \frac{9}{21} \left(1 - \frac{9}{21}\right) = 0.49$$

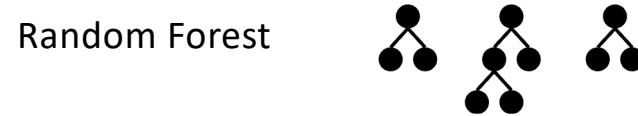
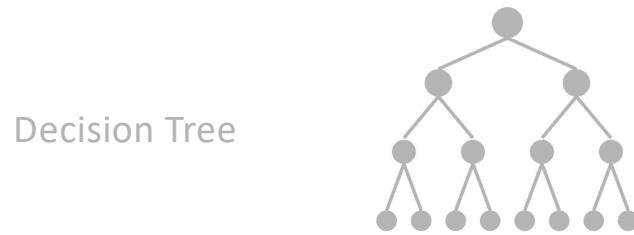
$$\frac{6}{14} \left(1 - \frac{6}{14}\right) + \frac{8}{14} \left(1 - \frac{8}{14}\right) = 0.49$$

$$\frac{6}{7} \left(1 - \frac{6}{7}\right) + \frac{1}{7} \left(1 - \frac{1}{7}\right) = 0.25$$

$$\text{Weighted Gini} = \frac{14}{21} 0.49 + \frac{7}{21} 0.25 = 0.41$$

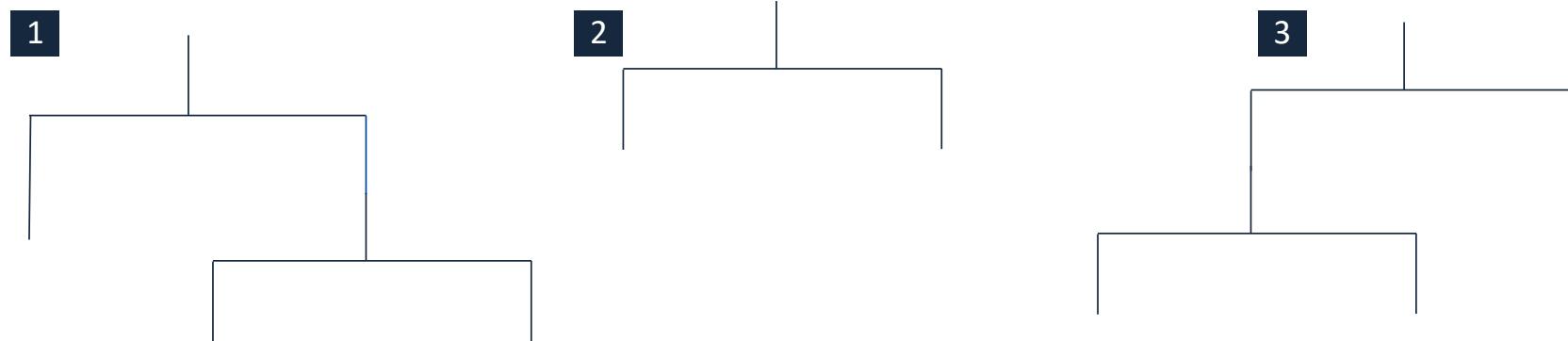
$$\text{Improvement} = 0.49 - 0.41 = 0.08$$

Ensemble Learning



Random Forest

Question: Why should we only build one tree?



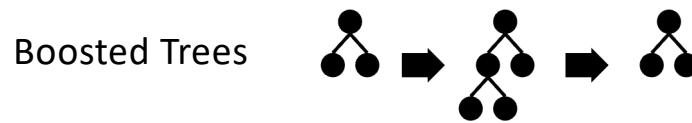
Used features:

Distance, weight, price

Customer

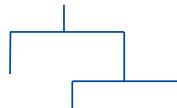
Customer, weight

Ensemble Learning



Gradient Boosted Trees

(1) Calculate 1. Tree



(2) Calculate prediction errors

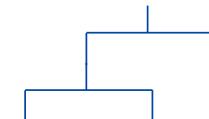
(3) Use Errors as target for 2. Tree



(4) Update predictions by calculating:

*Prediction 1. Tree + learning rate * Prediction 2. Tree*

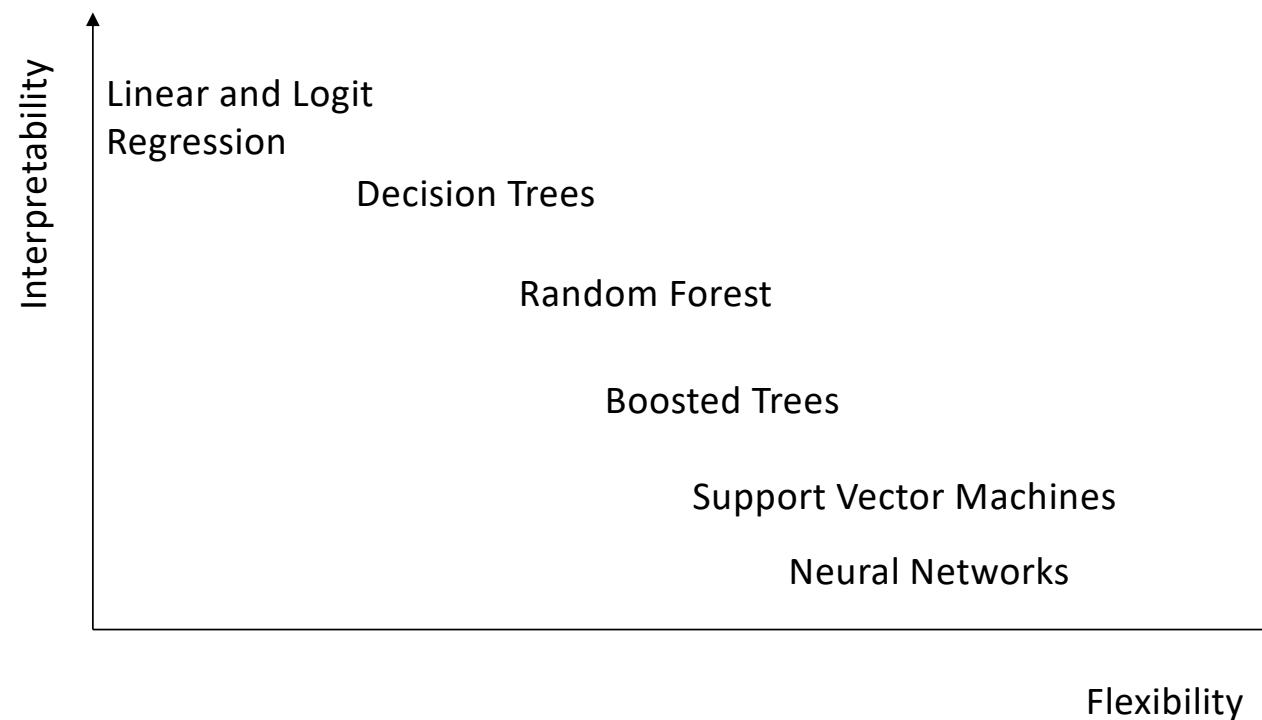
(5) Use updated errors as target for 3. Tree



(6) Update predictions by calculating:

*Prediction 1. Tree + learning rate * Prediction 2. Tree + learning rate * Prediction 3. Tree*

Interpretability Vs Flexibility Trade-off



Source James et al (2023), p. 24.

Agenda

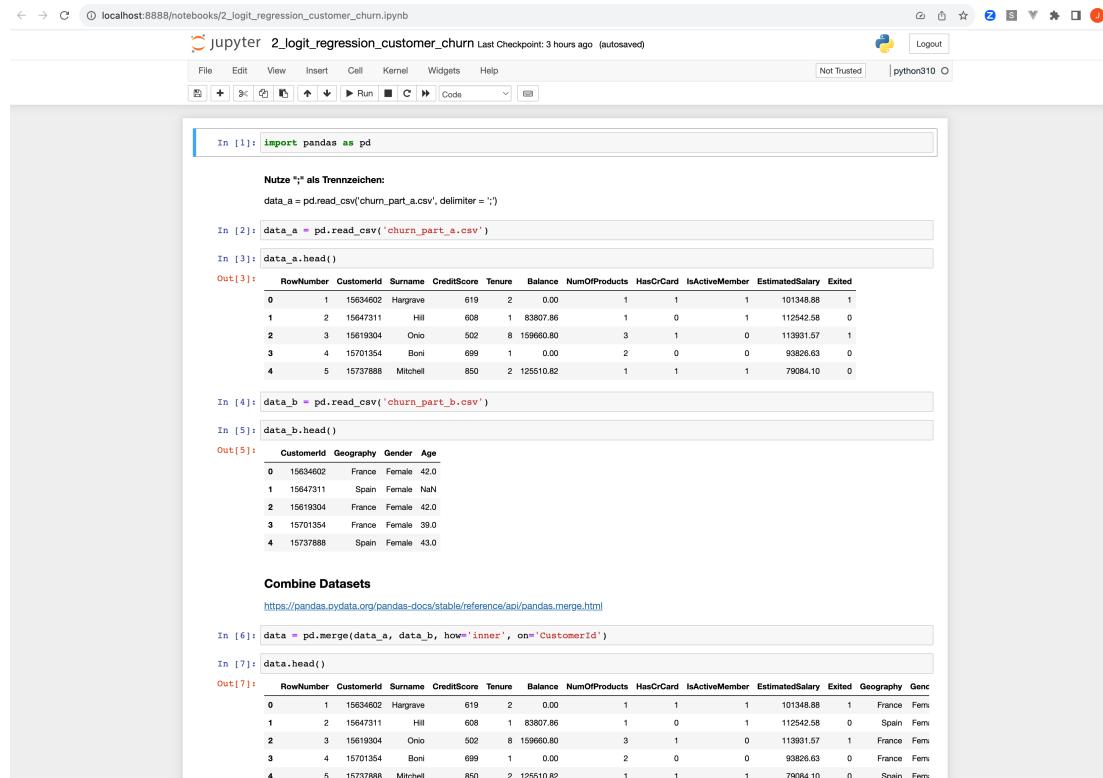
Business Analytics

- Data Science**
 - Classification vs. Regression
 - Prediction Models
- Case 3: Building a Prediction Model for Purchase Order Delivery
- Measuring the Quality of a Prediction Model
- Feature Engineering
- Explainable AI

Python - IDEs



Jupyter Notebooks



The screenshot shows a Jupyter Notebook interface with the following code and output:

```

In [1]: import pandas as pd
        Nutze ";" als Trennzeichen:
        data_a = pd.read_csv('churn_part_a.csv', delimiter = ';')

In [2]: data_a = pd.read_csv('churn_part_a.csv')

In [3]: data_a.head()
Out[3]:
   RowNumber CustomerId Surname CreditScore Tenure Balance NumOfProducts HasCrCard IsActiveMember EstimatedSalary Exited
0          1  15634602 Hargrave      619       2     0.00         1       1       1           1      101348.88      1
1          2  15647311    Hill       608       1   83867.86         1       0       1           1     112542.58      0
2          3  15619304     Onis       502       8  159660.80         3       1       0           0      113931.57      1
3          4  15701354     Boni       699       1     0.00         2       0       0           0      93826.63      0
4          5  15737888  Mitchell       850       2  125510.82         1       1       1           1      79084.10      0

In [4]: data_b = pd.read_csv('churn_part_b.csv')

In [5]: data_b.head()
Out[5]:
   CustomerId Geography Gender Age
0  15634602   France  Female  42.0
1  15647311    Spain  Female  NaN
2  15619304   France  Female  42.0
3  15701354   France  Female  39.0
4  15737888    Spain  Female  43.0

Combine Datasets
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.merge.html

In [6]: data = pd.merge(data_a, data_b, how='inner', on='CustomerId')

In [7]: data.head()
Out[7]:
   RowNumber CustomerId Surname CreditScore Tenure Balance NumOfProducts HasCrCard IsActiveMember EstimatedSalary Exited Geography Gender
0          1  15634602 Hargrave      619       2     0.00         1       1       1           1      101348.88      1   France  Fem
1          2  15647311    Hill       608       1   83867.86         1       0       1           1     112542.58      0   Spain  Fem
2          3  15619304     Onis       502       8  159660.80         3       1       0           0      113931.57      1   France  Fem
3          4  15701354     Boni       699       1     0.00         2       0       0           0      93826.63      0   France  Fem
4          5  15737888  Mitchell       850       2  125510.82         1       1       1           1      79084.10      0   Spain  Fem

```

Datastructures in Python

List: ["Müller AG", "Smith Ltd", "International Ltd"]

Dictionary: {"id": 1, "name": "Müller AG", "city": "Hamburg"}

List of Dictionaries: [
 {"id": 1, "name": "Müller AG", "city": "Hamburg"},
 {"id": 2, "name": "Smith Ltd", "city": "London"}]
]

Pandas Dataframe

	ID	Name	category
0	1	ElectraTech Supplies	Testing and Measurement Equipment
1	2	CircuitSolutions Inc.	Components and Parts Providers
2	3	PowerGrid Electronics	Electronics Manufacturing Tools and Equipment
3	4	Quantum Connectors	Connectivity and Communication Gear
4	5	MegaVolt Tech Distributors	Connectivity and Communication Gear

Getting Started with Pandas

```
# Importing pandas
import pandas as pd

# Reading an excel file
data = pd.read_excel('data.xlsx')

# Getting first 5 rows of data
data.head()

# Getting all column names
data.columns

# Get datatypes of column
data.dtypes
```

Agenda

Business Analytics

- Data Science**
 - Classification vs. Regression
 - Prediction Models
 - Case 3: Building a Prediction Model for Purchase Order Delivery
 - Measuring the Quality of a Prediction Model**
 - Feature Engineering
 - Explainable AI

Evaluating the Quality of a Model for a Regression Problem

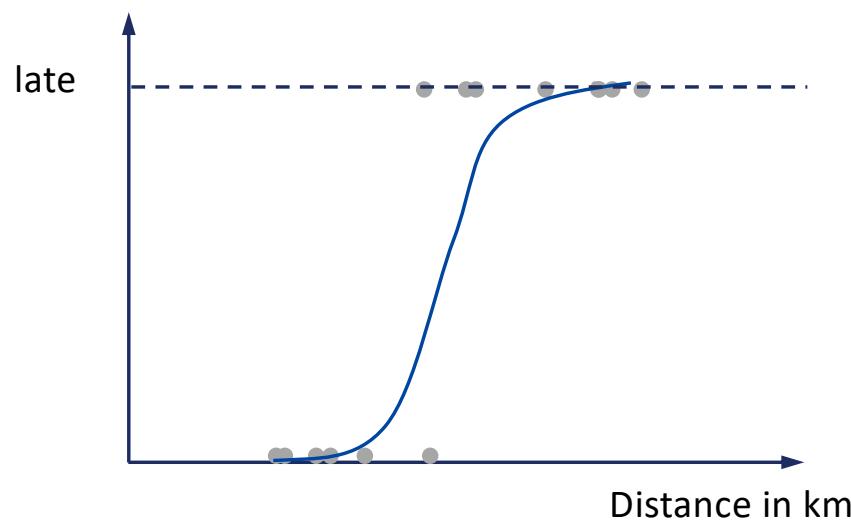
Observed (y_i)	Prediction (\hat{y}_i)
2	3
5	7
10	8

$$\text{Mean Squared Error} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$$\text{Mean Absolute Error} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Evaluating the Quality of a Model for a Classification Problem

Order	Confirmed Date	Distance in km	Actual Late	Prediction Model
#1	2024-06-14	502	True	90%
#2	2024-07-16	205	False	60%
#3	2024-07-14	275	False	33%



Confusion Matrix

		Actual condition	
		Positive (AP)	Negative (AN)
Predicted condition	Positive (PP)	True positive (TP)	False positive (FP)
	Negative (PN)	False negative (FN)	True negative (TN)

Confusion Matrix

Order	Probability (late)	Predicted (>=50%)	Actual	Actual condition		
				Predicted condition	Positive (AP)	Negative (AN)
#1	95%	Late	Late			
#2	90%	Late	In time	Positive (PP)		
#3	85%	Late	Late			
#4	80%	Late	Late	Negative (PN)		
#5	70%	Late	In time			
#6	60%	Late	Late			
#7	40%	In time	In time			
#8	30%	In time	In time			
#9	20%	In time	Late			
#10	10%	In time	In time			

Performance Measures Based on the Confusion Matrix

		Actual condition	
		Positive (AP)	Negative (AN)
Predicted condition	Positive (PP)	True positive (TP)	False positive (FP)
	Negative (PN)	False negative (FN)	True negative (TN)

Measure	Question	Formula
Accuracy	What percentage of the model's predictions were correct?	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	Out of all the positive predictions, how many were actually correct?	$\frac{TP}{TP + FP}$
True Positive Rate / Recall / Sensitivity	Of all the actual positive instances, how many did the model correctly identify?	$\frac{TP}{TP + FN}$
True Negative Rate / Specificity	Of all the actual negative instances, how many did the model correctly identify?	$\frac{TN}{TN + FP}$

Confusion Matrix

		Actual condition	
		Positive (AP)	Negative (AN)
Predicted condition	Positive (PP)	4 (TP)	2 (FP)
	Negative (PN)	1 (FN)	3 (TN)

Measure	Formula	
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN} =$	
Precision	$\frac{TP}{TP + FP} = \frac{4}{4 + 2} =$	
True Positive Rate / Recall / Sensitivity	$\frac{TP}{TP + FN} = \frac{4}{4 + 1} =$	
False Positive Rate / Specificity	$\frac{TN}{TN + FP} = \frac{3}{3 + 2} =$	

Confusion Matrix

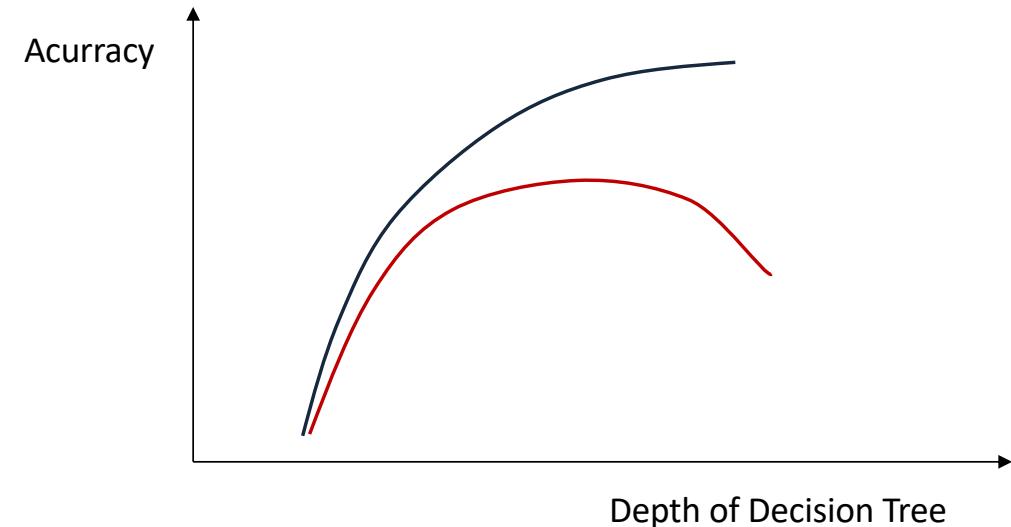
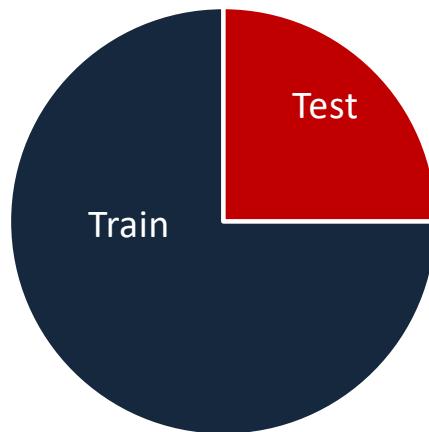
Order	Probability (late)	Predicted (>=80%)	Actual	Actual condition		
				Predicted condition	Positive (AP)	Negative (AN)
#1	95%	Late	Late	Positive (PP)	3 (TP)	1 (FP)
#2	90%	Late	In time	Negative (PN)	2 (FN)	4 (TN)
#3	85%	Late	Late			
#4	80%	Late	Late			
#5	70%	In time	In time			
#6	60%	In time	Late			
#7	40%	In time	In time			
#8	30%	In time	In time			
#9	20%	In time	Late			
#10	10%	In time	In time			

F-1 Score Is the Harmonic Mean Between Precision and Recall

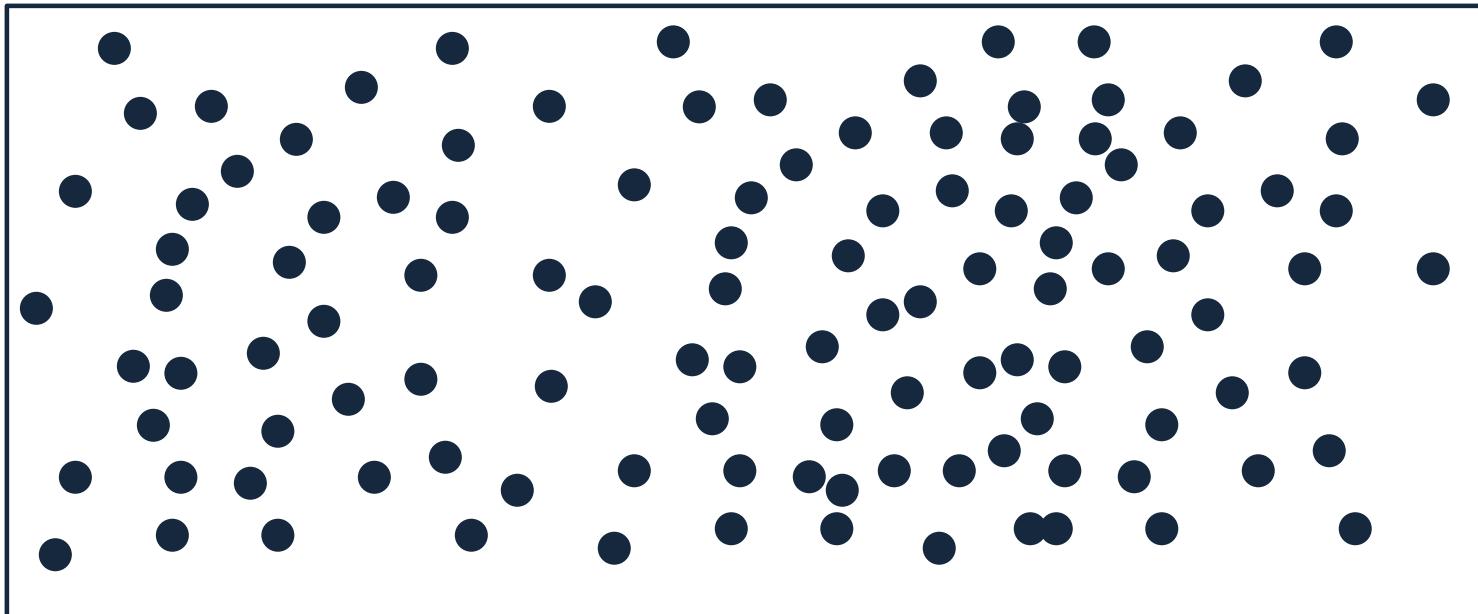
Threshold	Precision	Recall	F1 Score (Harmonic Mean of Precision and Recall)*
0.90	50%	20%	28.6%
0.80	75%	60%	66.7%
0.60	66.7%	80%	72.7%
0.50	66.7%	80%	72.7%
0.20	56%	100%	72.7%

$$* F1 - Score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

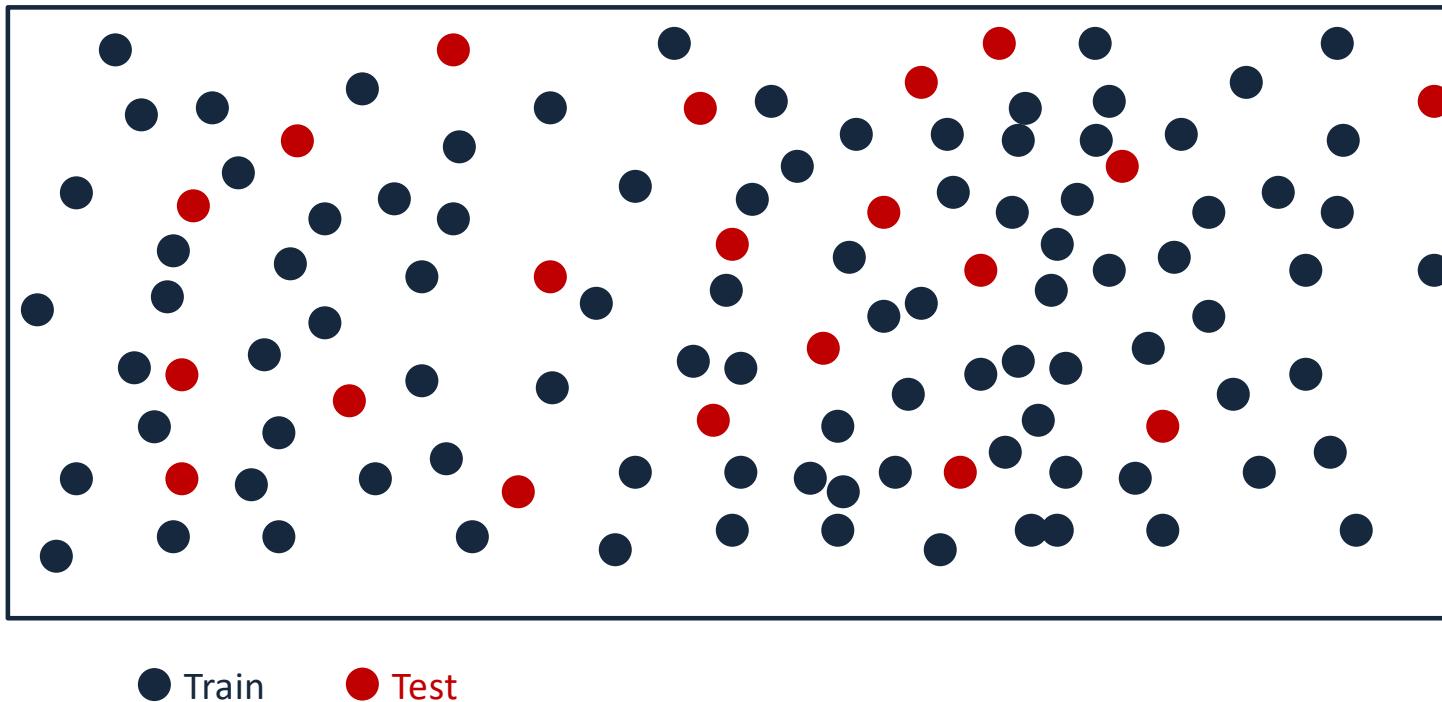
Train Vs. Test Data



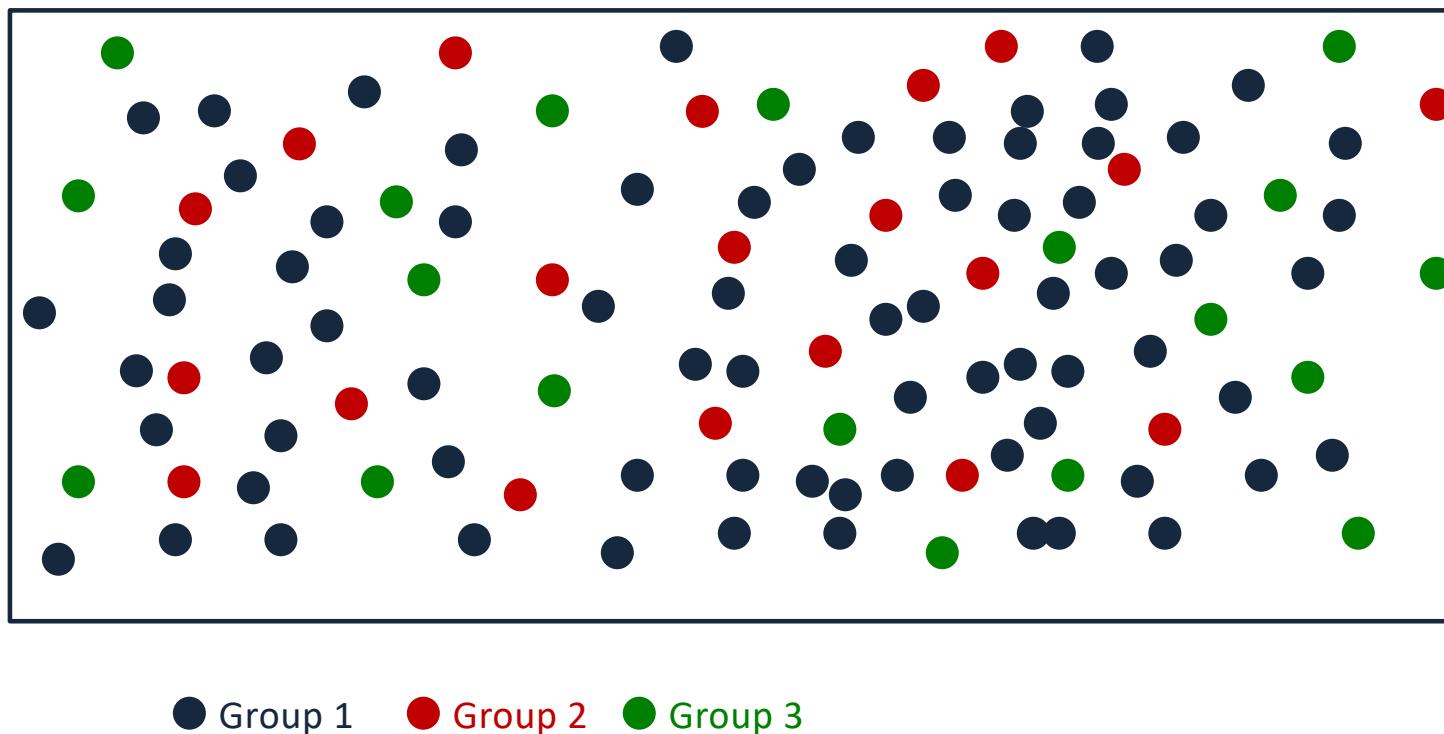
Train Vs. Test Data



Train Vs. Test Data

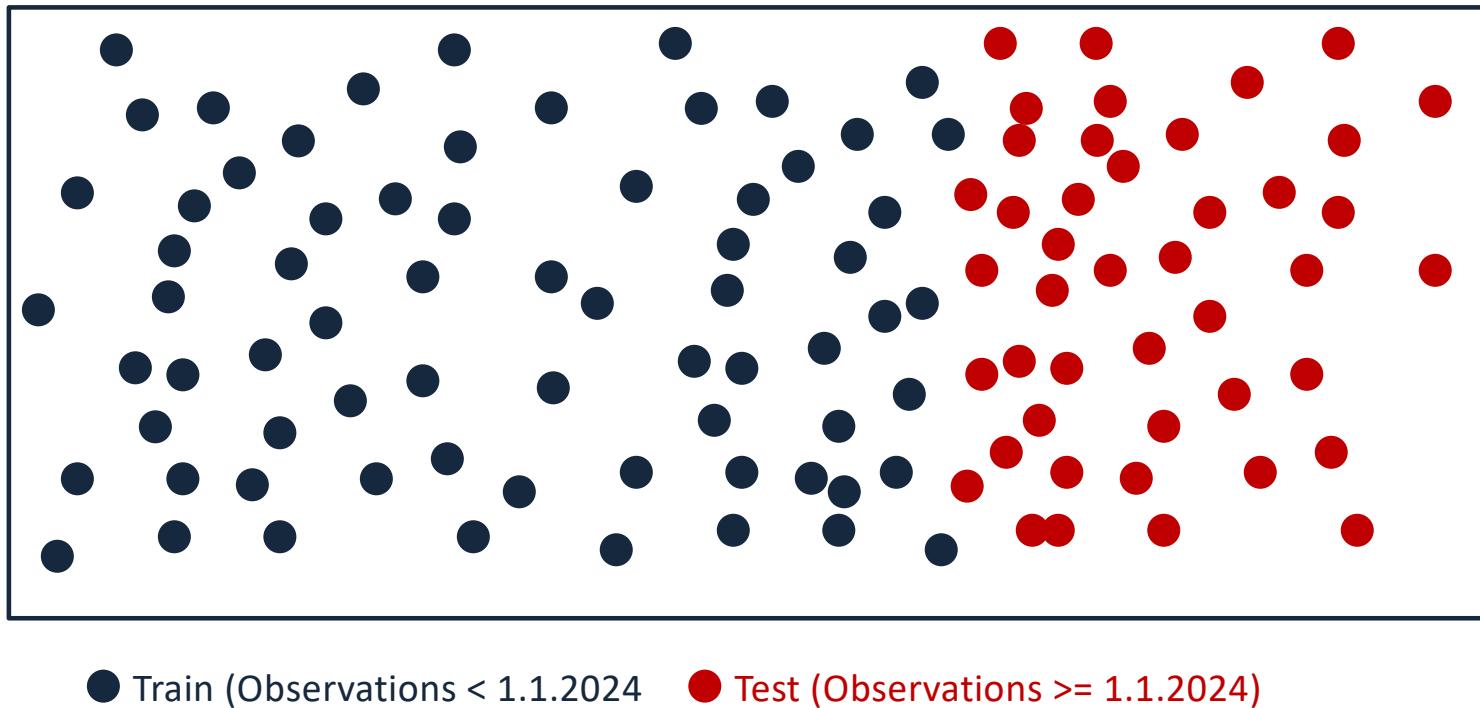


Cross-validation



Step	Train	Test
1	Group 1 & 2	Group 3
2	Group 1 & 3	Group 2
3	Group 2 & 3	Group 1

Splitting Time Series Data



Agenda

Business Analytics

Data Science

Classification vs. Regression

Prediction Models

Case 3: Building a Prediction Model for Purchase Order Delivery

Measuring the Quality of a Prediction Model

Feature Engineering

Explainable AI

Categorical Variables

Order	Customer Country
#1	France
#2	Germany
#3	Italy
#4	France



Order	Customer Country Numeric
#1	1
#2	2
#3	3
#4	1

Using One-hot Encoding for Categorical Variables

Order	Customer Country
#1	France
#2	Germany
#3	Italy
#4	France



Order	s_France	s_Germany	s_Italy
#1	1	0	0
#2	0	1	0
#3	0	0	1
#4	1	0	0

```
pandas.get_dummies(data["customer_state"], prefix="s")
```

Handling Date Information

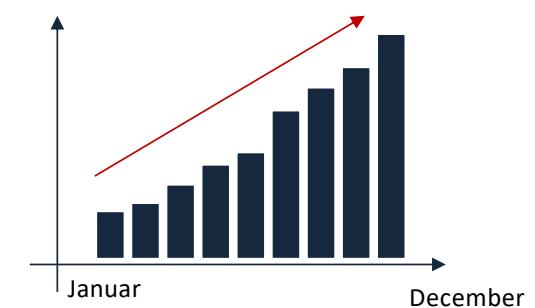
Order	Month
#1	August
#2	September
#3	December
#4	January



Option 1:

Order	Month Numeric
#1	8
#2	9
#3	12
#4	1

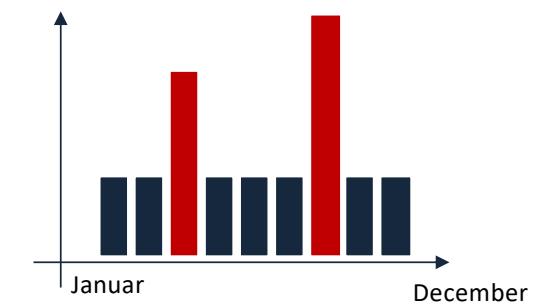
Late deliveries



Option 2:

Order	January	...	December
#1	0	...	0
#2	0	...	0
#3	0	...	1
#4	0	...	0

Late deliveries



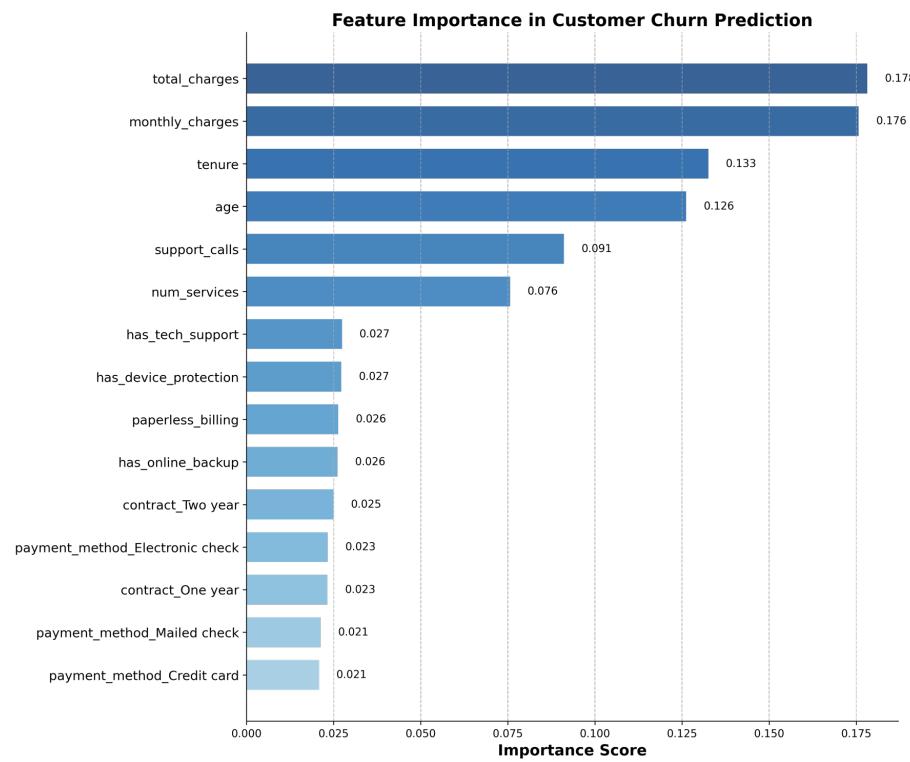
Agenda

Business Analytics

- Data Science**
 - Classification vs. Regression
 - Prediction Models
 - Case 3: Building a Prediction Model for Purchase Order Delivery
 - Measuring the Quality of a Prediction Model
 - Feature Engineering
 - Explainable AI

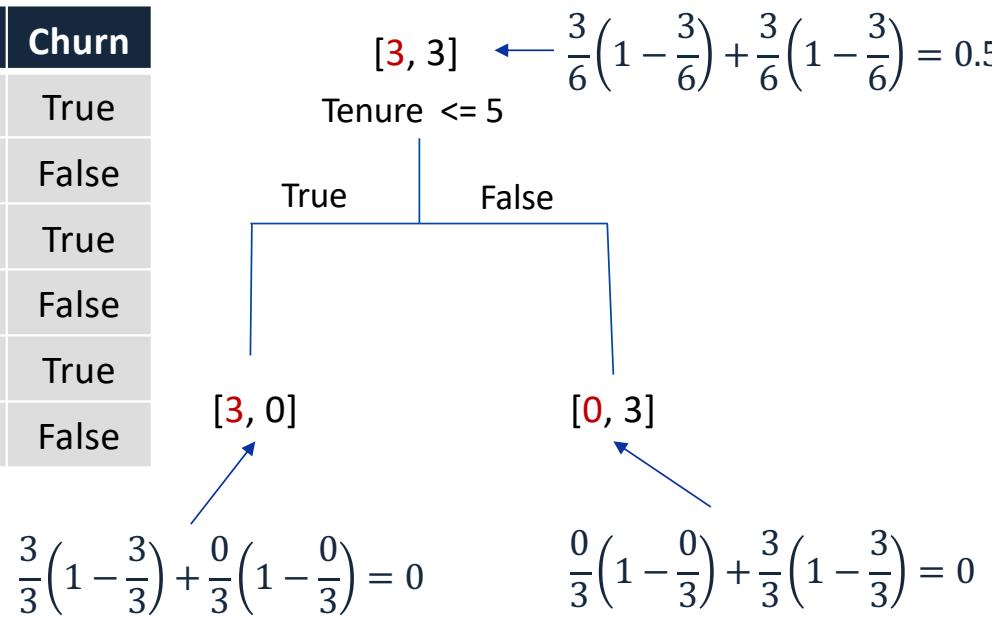
FEATURE IMPORTANCE

Feature Importance



Understanding Feature Importance

Age	Tenure	Churn
25	2	True
35	12	False
28	3	True
54	24	False
48	6	True
32	18	False

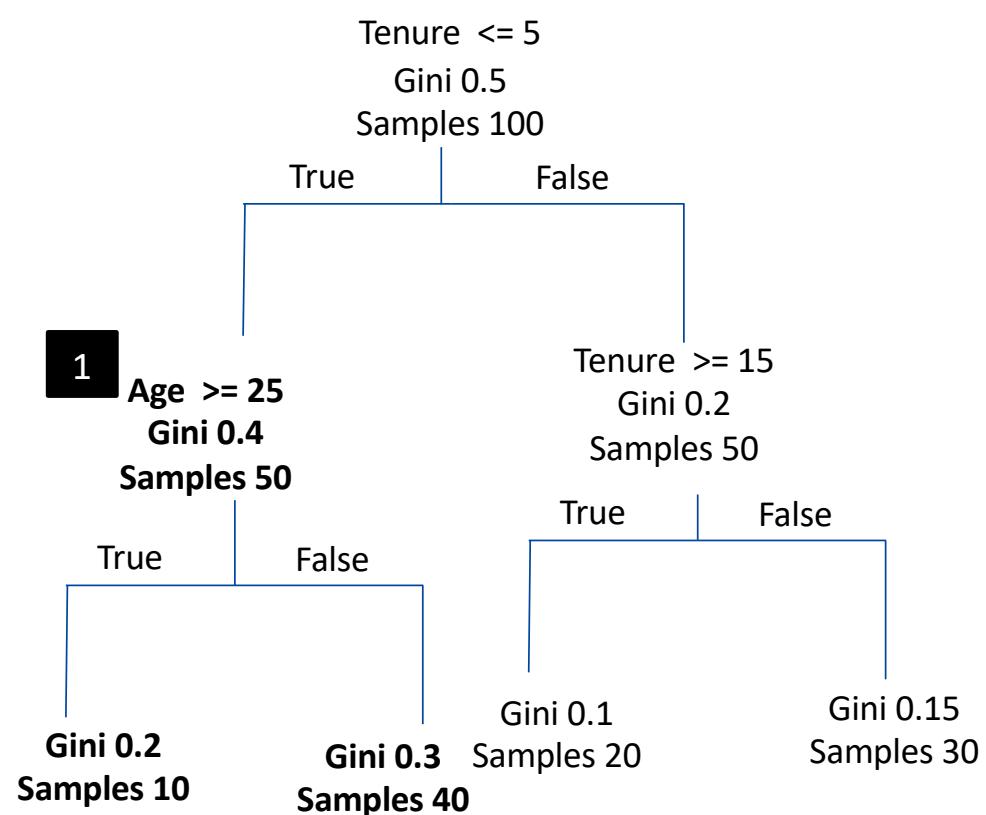


$$\text{Weighted Gini} = \frac{3}{6}0 + \frac{3}{6}0 = 0.0$$

$$\text{Improvement} = 0.5 - 0 = 0.5$$

- Repeat this for all nodes in which the features is used
- Weight the Improvement of each node with the samples in that node relative to the total samples

Understanding Feature Importance: Feature Age



Feature Age



1

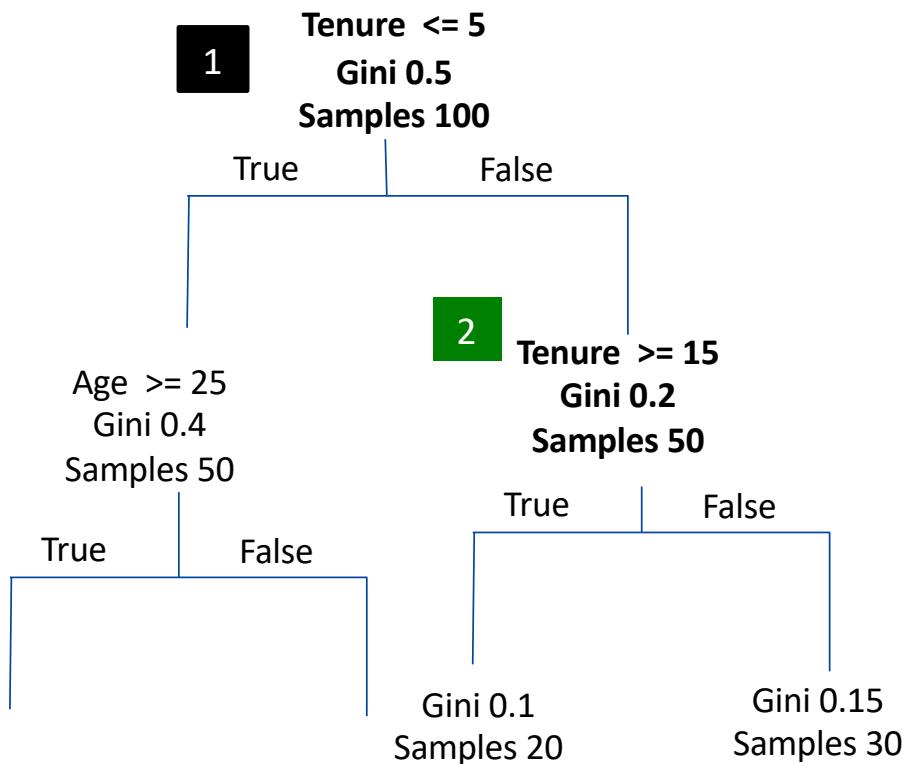
$$\text{Weighted Gini} = \frac{10}{50} 0.2 + \frac{40}{50} 0.3 = 0.28$$

$$\text{Improvement} = 0.4 - 0.28 = 0.12$$

$$\text{Weighted Improvement} = \frac{50}{100} 0.12 = 0.06$$

$$\text{Feature Importance} = 0.06 = 0.06$$

Understanding Feature Importance: Feature Tenure



1 $\text{Weighted Gini} = \frac{50}{100}0.4 + \frac{50}{100}0.2 = 0.3$

$\text{Improvement} = 0.5 - 0.3 = 0.2$

$\text{Weighted Improvement} = \frac{100}{100}0.2 = 0.2$

2 $\text{Weighted Gini} = \frac{20}{50}0.1 + \frac{30}{50}0.15 = 0.13$

$\text{Improvement} = 0.2 - 0.13 = 0.07$

$\text{Weighted Improvement} = \frac{50}{100}0.07 = 0.035$

Feature Importance = 0.2 + 0.035 = 0.235



SHAPLEY (SHAPLEY ADDITIVE EXPLANATIONS) VALUES

Shapley (shapley Additive Explanations) Values

Scenario:

Assume the following scenario: You've trained a machine learning model to predict apartment prices. For a certain apartment, it predicts €300,000, and you need to explain this prediction.

The apartment:

- has an area of 50 m,
- is located on the 2nd floor,
- has a park nearby,
- Cats are banned

The average prediction for all apartments is €310,000.

Our goal is to explain how each of these feature values contributed to the prediction. How much has each feature value contributed to the prediction compared to the average prediction?

Baseline	310.000
Area 50	- 10.000
2 nd floor	+ 1.500
Park nearby	+ 4.000
Cats are banned	- 4.500
Prediction	300.000

Source: Molnar, C. (2025)

Shapley (shapley Additive Explanations) Values: “cats Are Banned” Feature

Coalitions	Prediction (Cat banned)	Prediction (Cat not banned)	Difference
{}	310,000	312,000	-2,000
{park-nearby}	315,000	320,000	-5,000
{area-50}	325,000	330,000	-5,000
{floor-2nd}	312,000	317,000	-5,000
{park-nearby,area-50}	328,000	338,000	-10,000
{park-nearby,floor-2nd}	317,000	322,000	-5,000
{area-50,floor-2nd}	329,000	330,000	-1,000
{park-nearby,area-50,floor-2nd}	332,000	335,000	-3,000
Average			-4.500

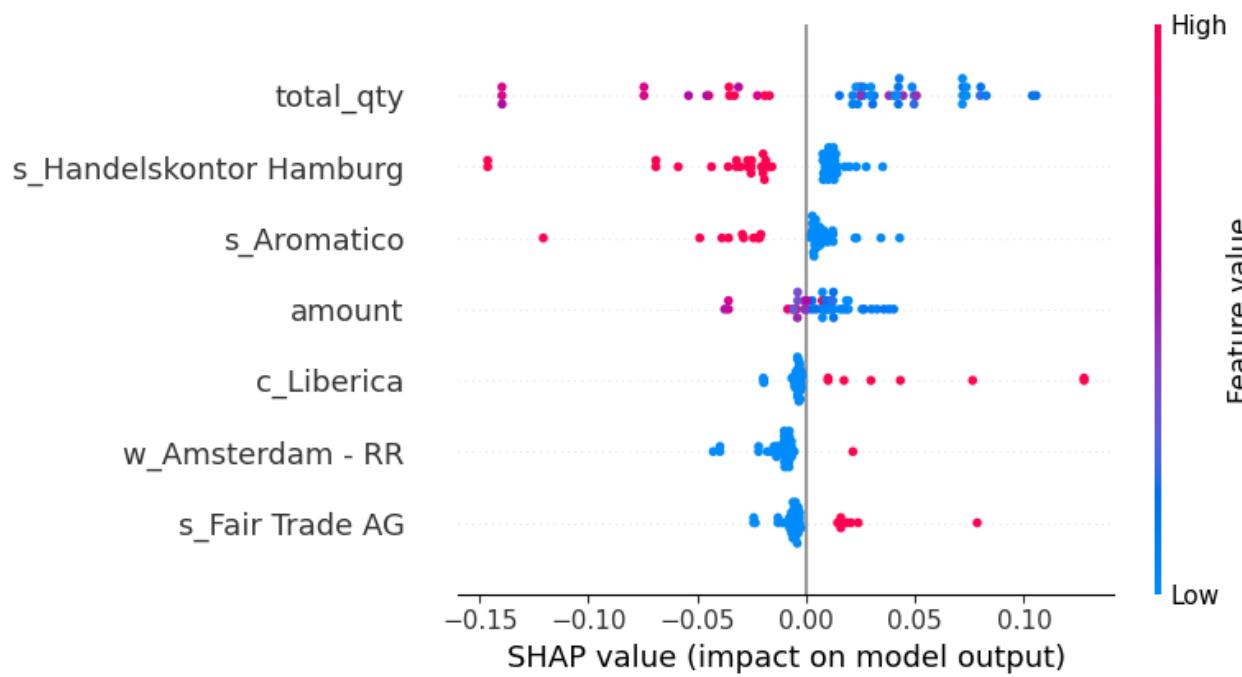


Shapley (shapley Additive Explanations) Values: “cats Are Banned” Feature

Coalitions	Prediction (Cat banned)	Prediction (Cat not banned)	Difference
{}	310,000	312,000	-2,000
{park-nearby}	315,000	320,000	-5,000
{area-50}	325,000	330,000	-5,000
{floor-2nd}	312,000	317,000	-5,000
{park-nearby,area-50}	328,000	338,000	-10,000
{park-nearby,floor-2nd}	317,000	322,000	-5,000
{area-50,floor-2nd}	329,000	330,000	-1,000
{park-nearby,area-50,floor-2nd}	332,000	335,000	-3,000
Average			-4.500



Shapley Values



Literature

- Date, C. J. (2011). SQL and relational theory: how to write accurate SQL code. "O'Reilly Media, Inc.".
- James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). Statistical learning. In *An introduction to statistical learning: With applications in Python*. Cham: Springer International Publishing.
- Kimball, R. (2013). The data warehouse toolkit: practical techniques for building dimensional data warehouses. John Wiley & Sons, Inc. 3rd edition.
- Kleppmann, M. (2017). Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems. O'Reilly Media, Inc.
- Linstedt, D., & Olschimke, M. (2015). Building a scalable data warehouse with data vault 2.0. Morgan Kaufmann.

Literature

- McKnight, W. (2013). Information management: strategies for gaining a competitive advantage with data. Newnes.
- McKinney, W. (2022). *Python for data analysis: Data wrangling with pandas, numpy, and jupyter.* " O'Reilly Media, Inc.".
- Molnar, C. (2025). *Interpretable Machine Learning: A Guide For Making Black Box Models Explainable.* (<https://christophm.github.io/interpretable-ml-book>)
- Simsion, G., & Witt, G. (2004). Data modeling essentials. Elsevier.
- Strengolt, P. (2023). Data Management at Scale. O'Reilly Media, Inc.



Master of Business Administration in Digital Business & AI
2024-2026

BACKUP

Querying Data

mytable

column1	column2	column3
a	b	c
x	q	t
v	w	y

```
SELECT column1, column2 FROM mytable WHERE column1 = 'x';
```

column1	column2
x	q

Querying Data

Select all rows and columns from a table

```
SELECT * FROM sales_order;
```

Select specific columns

```
SELECT name, customer FROM sales_order;
```

Get all distinct values from a column

```
SELECT DISTINCT name from sales_order;
```

Get the number of rows in a table

```
SELECT COUNT(*) from sales_order;
```

Sql Comparison Operators to Select Specific Rows

```
SELECT * FROM sales_order WHERE customer = 'Fantastic Coffee';
```

```
SELECT * FROM sales_order WHERE delivery_date > '2023-09-01';
```

Operator	Description
=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
<>	Not equal to

Sql Logical Operators

Operator	Description
ALL	TRUE if all of the subquery values meet the condition
AND	TRUE if all the conditions separated by AND is TRUE
ANY	TRUE if any of the subquery values meet the condition
BETWEEN	TRUE if the operand is within the range of comparisons
EXISTS	TRUE if the subquery returns one or more records
IN	TRUE if the operand is equal to one of a list of expressions
LIKE	TRUE if the operand matches a pattern
NOT	Displays a record if the condition(s) is NOT TRUE
OR	TRUE if any of the conditions separated by OR is TRUE
SOME	TRUE if any of the subquery values meet the condition

Sql Operators to Select Specific Rows

```
SELECT * FROM sales_order WHERE delivery_date > '2023-09-01' AND  
customer = 'Fantastic Coffee';
```

```
SELECT COUNT(*)  
FROM sales_order  
WHERE (delivery_date > '2023-09-01' AND customer = 'Fantastic Coffee')  
OR  
(delivery_date > '2023-09-01' AND customer='Cuppa Joe');
```

```
SELECT COUNT(*)  
FROM sales_order  
WHERE delivery_date > '2023-09-01' AND customer in ('Fantastic  
Coffee', 'Cuppa Joe');
```

Left Join: Using All Rows From the First Table

```
SELECT * FROM orders
LEFT JOIN customers ON orders.customer_id = customers.id;
```

order_id	customer_id	product	quantity	id	name	city	email	contact
1	1	Apples	10	1	Smith	New York	smit@...	07979
2	2	Bananas	20	2	Doe	Frankfurt	doe@...	2347
3	4	Cherries	30	NULL	NULL	NULL	NULL	NULL
4	1	Dates	40	1	Smith	New York	smit@...	07979

Pandas equivalent:

```
pd.merge(customers, suppliers, how='left', left_on='customer_id', right_on='id')
```

Right Join: Using All Rows From the Second Table

```
SELECT * FROM orders
RIGHT JOIN customers ON orders.customer_id = customers.id;
```

order_id	customer_id	product	quantity	id	name	city	email	contact
1	1	Apples	10	1	Smith	New York	smit@...	07979
2	2	Bananas	20	2	Doe	Frankfurt	doe@...	2347
4	1	Dates	40	1	Smith	New York	smit@...	07979
NULL	NULL	NULL	NULL	3	Wang	Munich	wang@	143

Pandas equivalent:

```
pd.merge(customers, suppliers, how='right', left_on='customer_id', right_on='id')
```

Outer Join: Using All Rows From Both Tables

```
SELECT * FROM orders
INNER JOIN customers ON orders.customer_id = customers.id;
```

order_id	customer_id	product	quantity	id	name	city	email	contact
1	1	Apples	10	1	Smith	New York	smit@...	07979
2	2	Bananas	20	2	Doe	Frankfurt	doe@...	2347
4	1	Dates	40	1	Smith	New York	smit@...	07979

Pandas equivalent:

```
pd.merge(customers, suppliers, how='inner', left_on='customer_id', right_on='id')
```

Sql Operators to Select Specific Rows

```
SELECT * FROM orders
FULL OUTER JOIN customers ON orders.customer_id = customers.id;
```

order_id	customer_id	product	quantity	id	name	city	email	contact
1	1	Apples	10	1	Smith	New York	smit@...	07979
2	2	Bananas	20	2	Doe	Frankfurt	doe@...	2347
3	4	Cherries	30	NULL	NULL	NULL	NULL	NULL
4	1	Dates	40	1	Smith	New York	smit@...	07979
NULL	NULL	NULL	NULL	3	Wang	Munich	wang@	143

Pandas equivalent:

```
pd.merge(customers, suppliers, how='outer', left_on='customer_id', right_on='id')
```