UNIVERSITY of DEBRECEN

Peter Lengyel

István Füzesi

Supporting Business Analysis with Artificial Intelligence in Power BI

Interreg

Co-funded by the European Union

Romania - Hungary

HUNGARIAN GOVERNMENT

Project title: „**Exchange of Experience for research and usage of Artificial Intelligence' Advanced Techniques in finance, accounting and business administration in Bihor-Hajdu Bihar Region**”, **Acronim: E2U-AI**

Project code: ***ROHU00120,*** Subsidy contract no. 37950/19.02.2025

Course Schedule

Day	Program	Format
Monday	Introduction to Power BI with practical exercises and report creation	In-person
Tuesday	Exploring Business Intelligence, databases, and AI topics	Online
Wednesday	Introduction to KNIME	Online
Thursday	Studying Power BI examples, exploring AI tools	Online
Friday	Solving practical KNIME exercises	In-person

AI Tools Used in the Training

- The training 'Supporting Business Analysis with Artificial Intelligence in Power BI' utilised paid ChatGPT and Claude Sonnet AI.
- Paid versions enabled advanced reasoning capabilities, essential for complex business analysis tasks.
- Higher token limits and faster response times ensured efficient workflow during the course.

Why Premium AI Tools?

- Premium models provided more accurate insights for Power BI use cases.
- Enhanced data interpretation improved the quality of analytical outputs.
- Consistent performance and reliability were key for demonstrating AI-driven business analysis.

Mastering
Data Visualization
with

What is Power BI

Power BI is Microsoft's business analytics platform that helps you turn data into actionable insights. Whether you're a business user, report creator, or developer, Power BI offers integrated tools and services to connect, visualize, and share data across your organization.

Power BI: Step-by-step

- Get started: Sign up and set up your workspace. Download Power BI Desktop or use the Power BI service in your browser.
- Connect and prepare data: Connect to sources like Excel, SQL, or cloud services. Clean and shape your data.
- Model and combine data: Create relationships, add calculations, and combine sources for a complete view.
- Build reports and dashboards: Use drag-and-drop tools to create interactive visuals.
- Explore and analyze: Filter, sort, and drill down to find insights. Use built-in analytics.
- Share and collaborate: Publish to the Power BI service, share with your team, and collaborate in real time.
- Administer and secure: Manage access, set up security roles, and monitor usage.

Power BI Desktop versus the Power BI service

- Power BI has two main components: Power BI Desktop and the Power BI service. Desktop is best for data modeling and report creation, while the service is ideal for sharing and collaboration. Both can connect to data sources and create visualizations. There's also a Power BI Mobile app for viewing reports on the go.

Need	Use this	Why
Create reports	Power BI Desktop	Full data modeling and design tools
Share with team	Power BI service	Collaboration and sharing features
View on mobile	Power BI Mobile apps	Optimized for phones and tablets

Core features of Power BI Desktop and the Power BI service

Power BI Desktop:

- Connect to 100+ data sources (databases, cloud, files, web)
- Power Query Editor for data transformation
- Data modeling with DAX, calculated columns, and relationships
- 30+ built-in and custom visuals
- Advanced features: performance analyzer, external tools, composite models

Power BI Service:

- Workspaces for team collaboration
- Apps for distributing dashboards and reports
- Dataflows for reusable data prep
- Datasets shared across reports
- Real-time dashboards and streaming data
- Schedule refresh, email subscriptions, alerts, Q&A, embedding, export
- Security: RLS, sensitivity labels, usage metrics, audit logs

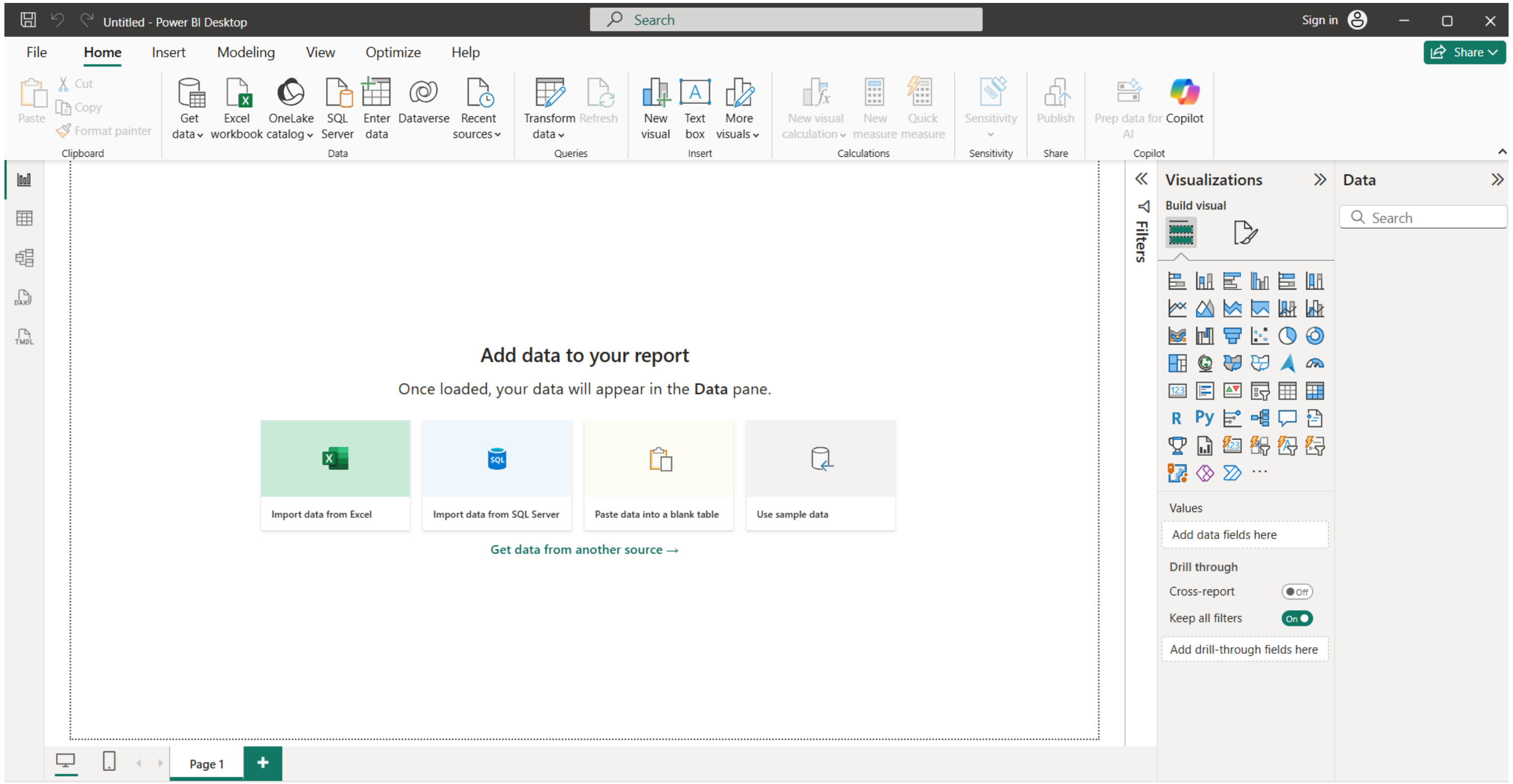
What is Power BI Desktop?

Power BI Desktop is a free Windows application that lets you connect to data, transform it, and create interactive visual reports. It’s part of the Power BI suite, which also includes the Power BI service (online) where you can publish and share your reports.

With Power BI Desktop, you can:

- Import data from Excel, databases, web sources, and more
- Clean and shape your data using Power Query
- Build visuals like charts, maps, and tables
- Publish your reports to the Power BI service for sharing

Explore the Power BI Desktop interface

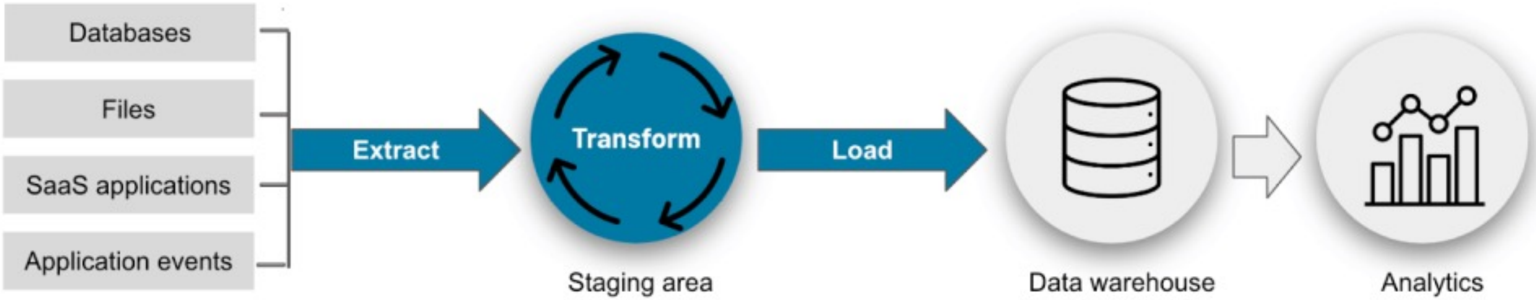


Add data to your report

Once loaded, your data will appear in the **Data** pane.

What is ETL process?

ETL refers to the process of transferring data from source to destination warehouse. It is an acronym for Extract, Transform, and Load.



Extract

In this first step, data is extracted from various sources and all the different formats and collected in the staging area.

- *Full Extraction*: In this method, data is completely extracted from the source system.
- *Update Modification*: Many ETL tools or databases have a mechanism where the system notifies you when the record or the data has been changed.
- *Incremental Extraction*: In incremental extraction, the changes in source data need to be tracked since the last successful extraction. Only these changes in data will be retrieved and loaded.

Data Transformation

There are several sub-processes involved in Data Transformation:

- **Cleaning**: Mapping of NULLs to 0 or “Male” to “M” and “Female” to “F,” date format consistency, etc.
- **Filtering**: Only selective numbers of rows and columns are chosen as required for further analysis.
- **Deduplication**: Identification and removal of any duplicate records.
- **Derivation**: Using your data to derive new value from existing data.
- **Format Revision**: Conversion of data/time format, character set, etc.
- **Joining**: Linking of data using a set of predefined rules.
- **Integration**: Reconciling different data names and values for the same data element.
- **Verification**: Unused/superfluous Data is removed.

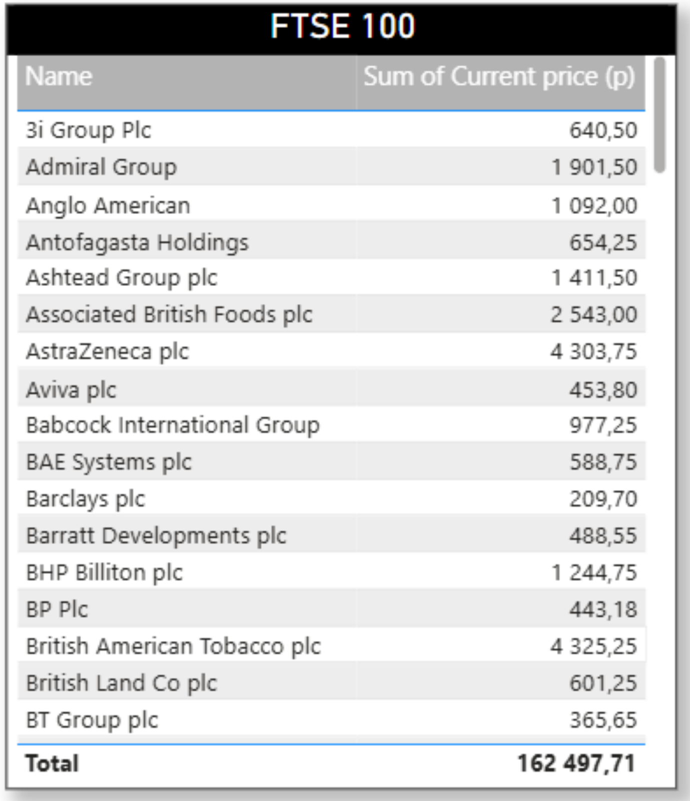
Load

The last step is where you successfully load the data into its new destination.

- **Full Load**: Full Load in ETL is loading all the data from the source to the destination.
- **Incremental Load**: In case the amount of data is too large for it to be loaded in one go or where the actual updated records are very less but the whole data is very huge, we go with the incremental load.

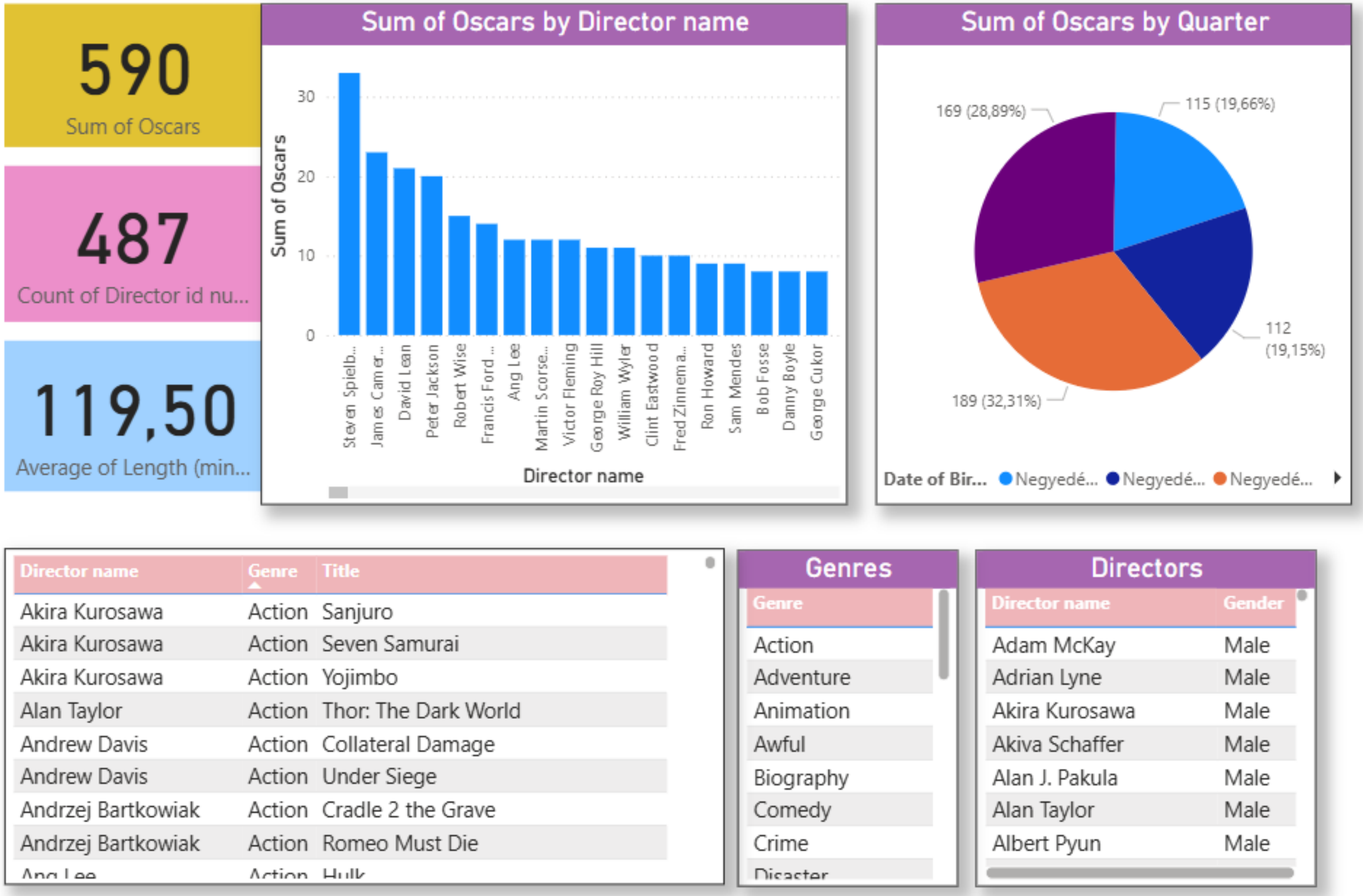
Power BI Exercises

The top 1,000 shares in the FTSE index, as imported from the Hargreaves Lansdown website



FTSE 100

Name	Sum of Current price (p)
3i Group Plc	640,50
Admiral Group	1 901,50
Anglo American	1 092,00
Antofagasta Holdings	654,25
Ashtead Group plc	1 411,50
Associated British Foods plc	2 543,00
AstraZeneca plc	4 303,75
Aviva plc	453,80
Babcock International Group	977,25
BAE Systems plc	588,75
Barclays plc	209,70
Barratt Developments plc	488,55
BHP Billiton plc	1 244,75
BP Plc	443,18
British American Tobacco plc	4 325,25
British Land Co plc	601,25
BT Group plc	365,65
Total	**162 497,71**



590
Sum of Oscars

487
Count of Director id nu...

119,50
Average of Length (min...

Director name	Genre	Title
Akira Kurosawa	Action	Sanjuro
Akira Kurosawa	Action	Seven Samurai
Akira Kurosawa	Action	Yojimbo
Alan Taylor	Action	Thor: The Dark World
Andrew Davis	Action	Collateral Damage
Andrew Davis	Action	Under Siege
Andrzej Bartkowiak	Action	Cradle 2 the Grave
Andrzej Bartkowiak	Action	Romeo Must Die
Ang Lee	Action	Hulk

Genres

Genre
Action
Adventure
Animation
Awful
Biography
Comedy
Crime
Disaster

Directors

Director name	Gender
Adam McKay	Male
Adrian Lyne	Male
Akira Kurosawa	Male
Akiva Schaffer	Male
Alan J. Pakula	Male
Alan Taylor	Male
Albert Pyun	Male

Games of Thrones Viewers

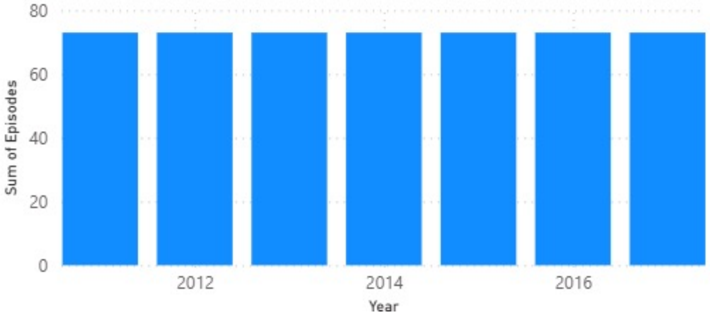
Season	Year	Average of U.S. viewers (millions)
1	2011	2,52
2	2012	3,80
3	2013	4,97
4	2014	6,85
5	2015	6,88
6	2016	7,69
7	2017	10,26
Total		**5,95**

398,73

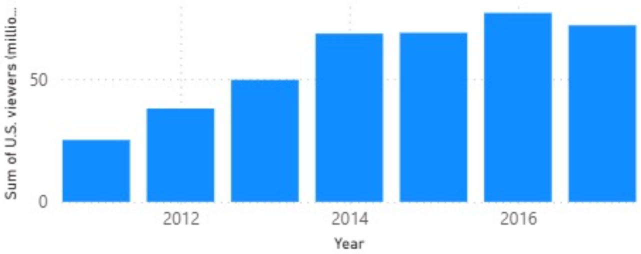
Sum of U.S. viewers (millions)

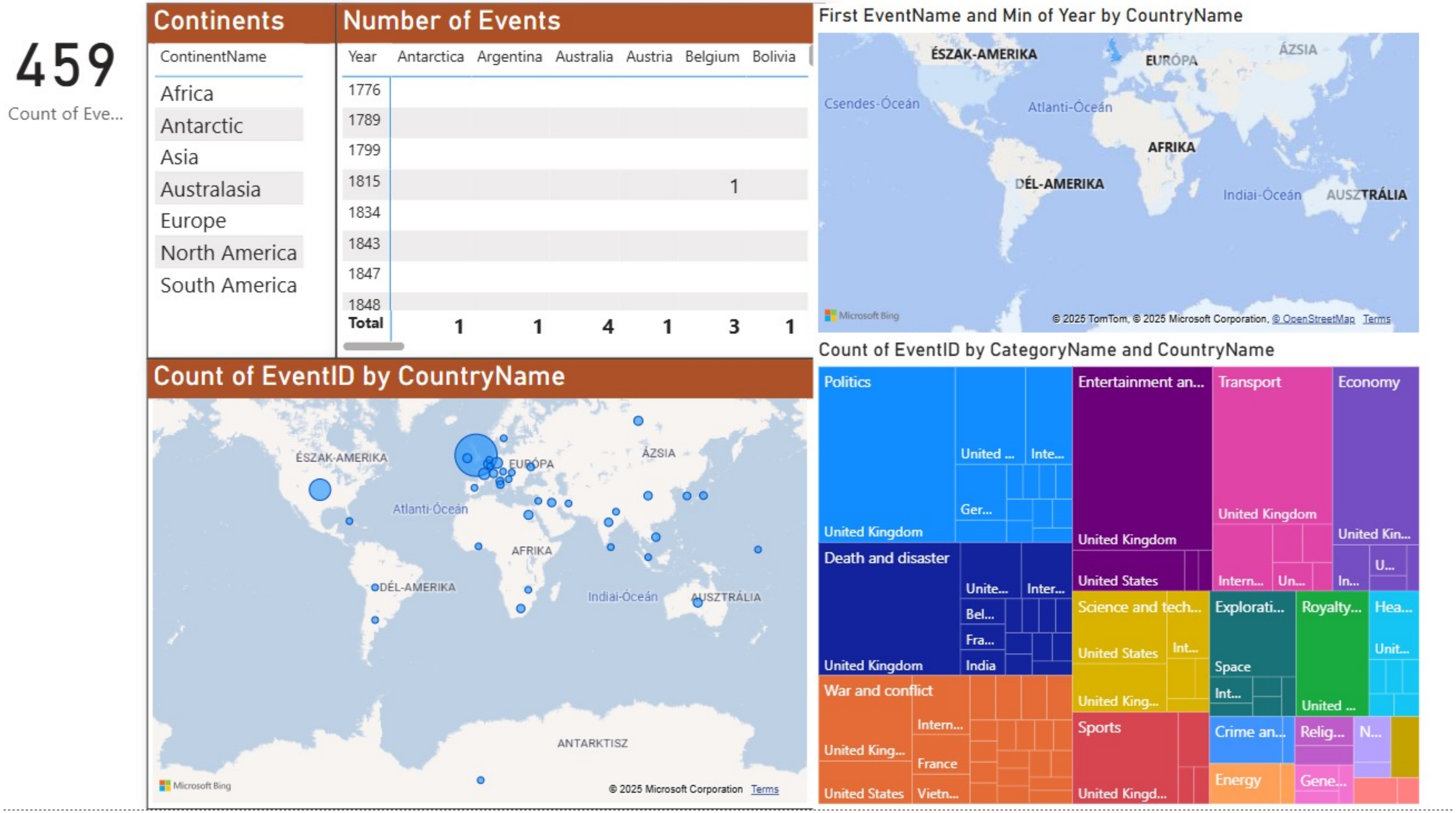
GAME OF THRONES
WINTER IS COMING™
The officially licensed browser game
YOOZOO GAMES
MIRACLE GAMES
WB GAMES
HBO HOME BOX OFFICE
Game of Thrones series title and artwork © 2019 Home Box Office, Inc. All Rights Reserved.
HBO and related trademarks are the property of Home Box Office, Inc. Under license to WB Games.

Sum of Episodes by Year



Sum of U.S. viewers (millions) by Year





459

Count of Eve...

Continents

ContinentName

- Africa
- Antarctic
- Asia
- Australasia
- Europe
- North America
- South America

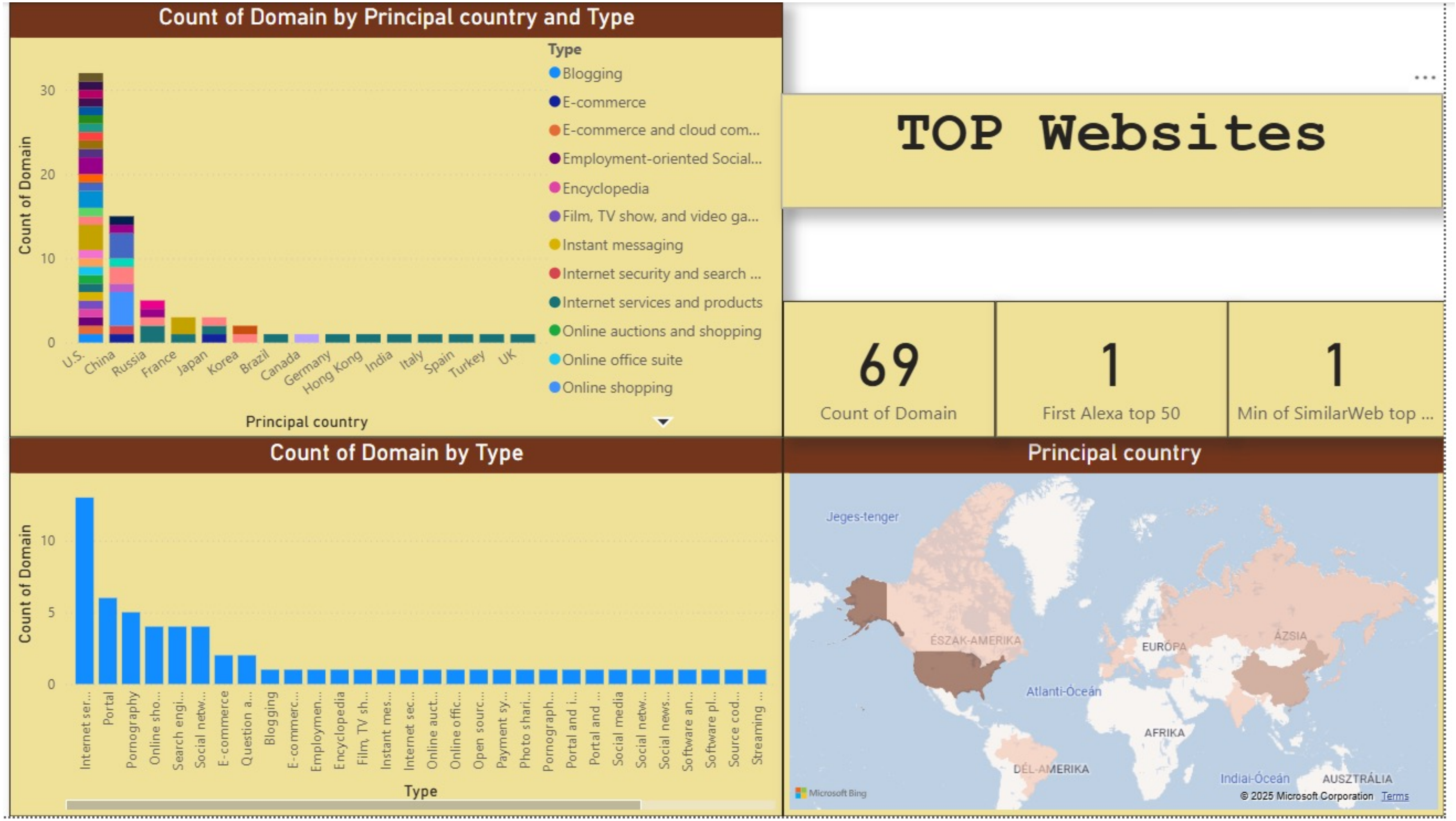
Number of Events

Year	Antarctica	Argentina	Australia	Austria	Belgium	Bolivia
1776						
1789						
1799						
1815					1	
1834						
1843						
1847						
1848						
Total	**1**	**1**	**4**	**1**	**3**	**1**

First EventName and Min of Year by CountryName

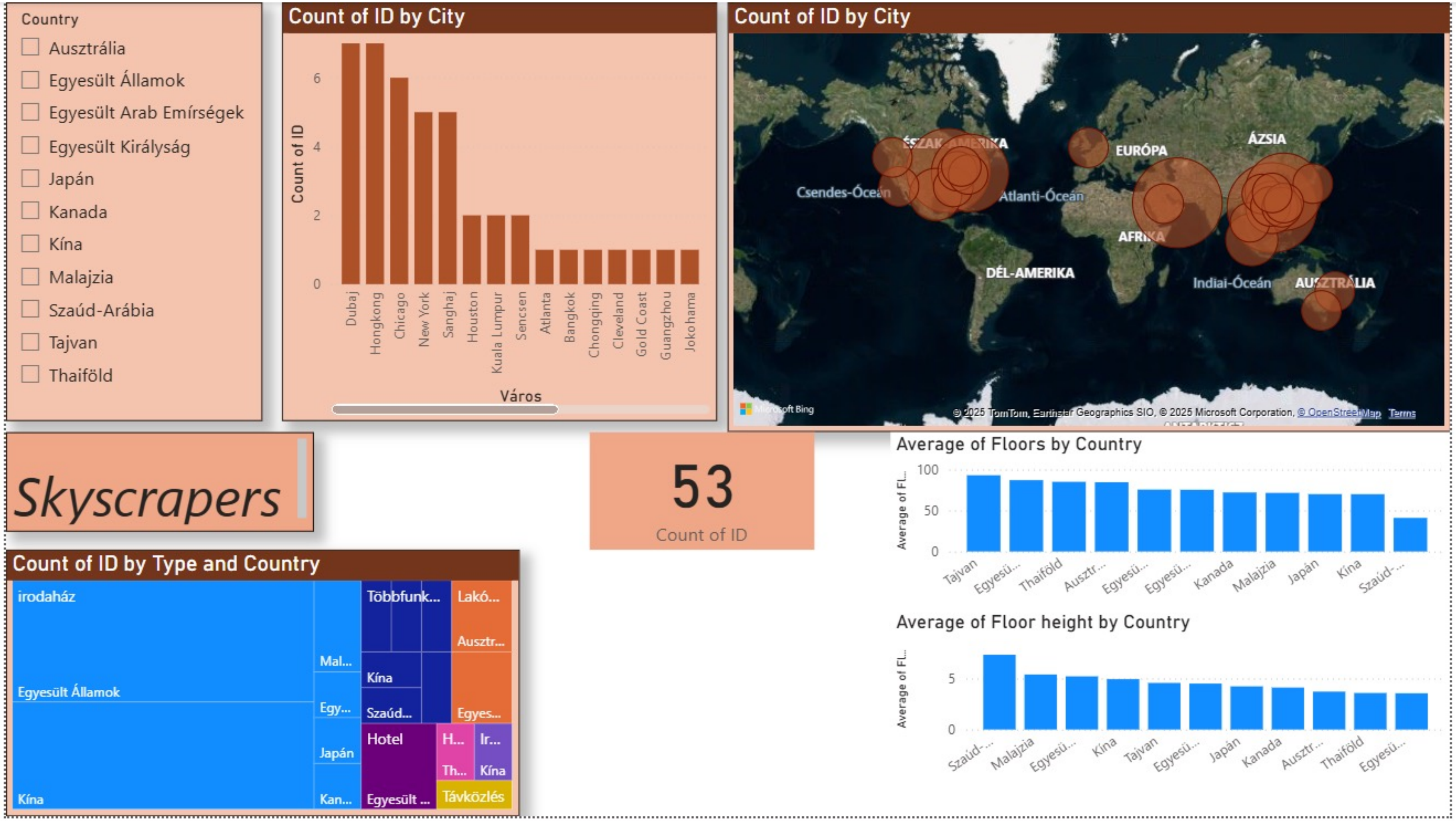
Count of EventID by CountryName

Count of EventID by CategoryName and CountryName



TOP Websites

69	1	1
Count of Domain	First Alexa top 50	Min of SimilarWeb top ...



Country

- □ Ausztrália
- □ Egyesült Államok
- □ Egyesült Arab Emírségek
- □ Egyesült Királyság
- □ Japán
- □ Kanada
- □ Kína
- □ Malajzia
- □ Szaúd-Arábia
- □ Tajvan
- □ Thaiföld

Count of ID by City

Count of ID by City

Skyscrapers

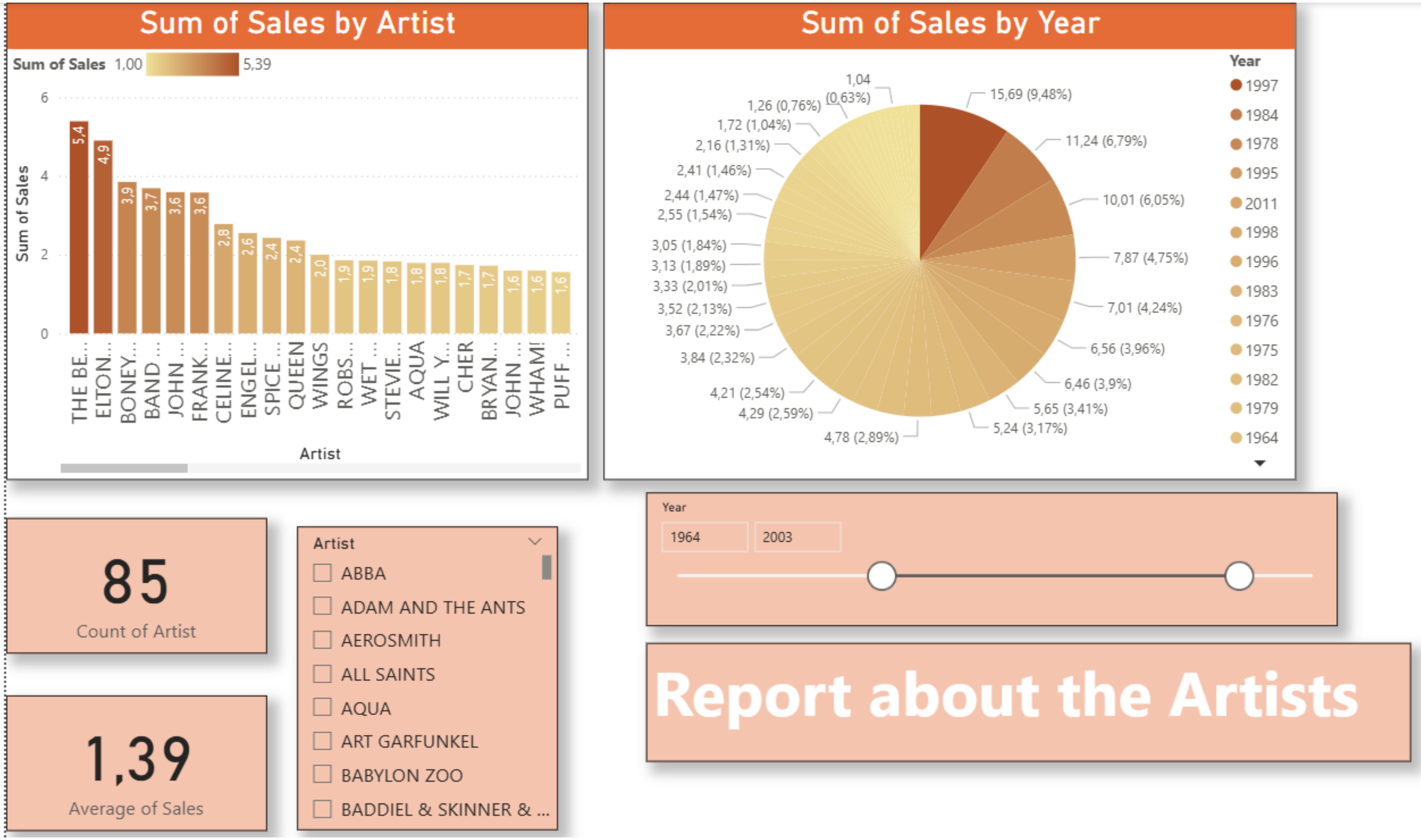
53

Count of ID

Average of Floors by Country

Count of ID by Type and Country

Average of Floor height by Country



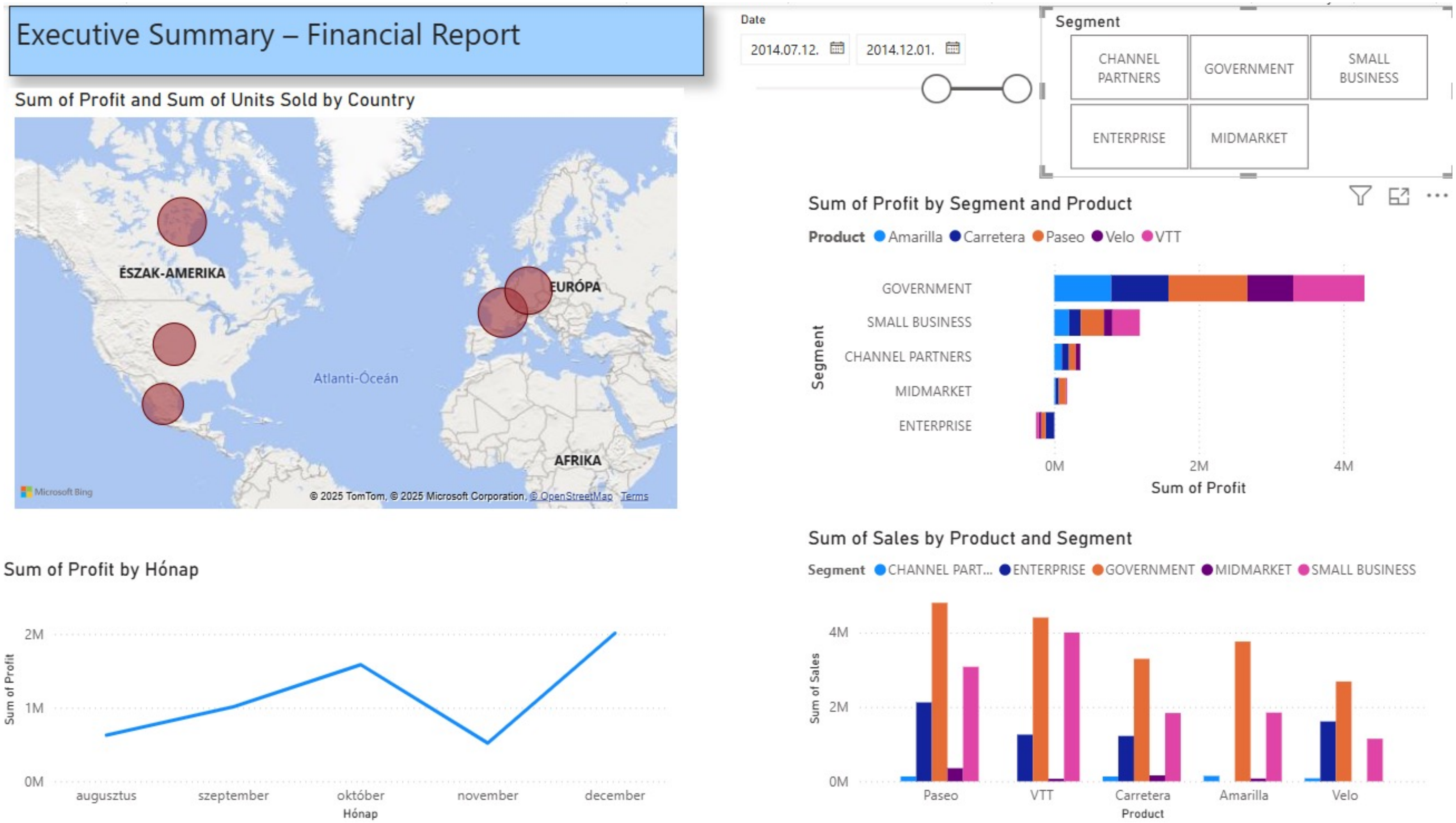
85
Count of Artist

1,39
Average of Sales

Artist
□ ABBA
□ ADAM AND THE ANTS
□ AEROSMITH
□ ALL SAINTS
□ AQUA
□ ART GARFUNKEL
□ BABYLON ZOO
□ BADDIEL & SKINNER & ...

Year
1964 2003

Report about the Artists



Executive Summary – Financial Report

Date

2014.07.12.

2014.12.01.

Segment

CHANNEL PARTNERS

GOVERNMENT

SMALL BUSINESS

ENTERPRISE

MIDMARKET

Sum of Profit and Sum of Units Sold by Country

ÉSZAK-AMERIKA
EURÓPA
Atlanti-Óceán
AFRIKA
Microsoft Bing
© 2025 TomTom, © 2025 Microsoft Corporation, © OpenStreetMap Terms

Sum of Profit by Segment and Product

Sum of Profit by Hónap

Sum of Sales by Product and Segment

Database Systems and Business Intelligence

Data Management

Without data and the ability to process the data:

- An organization could not successfully complete most business activities

Data consists of raw facts.

To transform data into useful information:

- It must first be organized in a meaningful way

Database Management System (DBMS):

A group of programs that manipulate the database and provide an interface between the database and the user of the database and other application programs

Database Administrator (DBA):

A skilled IS professional who directs all activities related to an organization's database.

The Hierarchy of Data

Bit (a binary digit):

- Circuit that is either on or off

Byte:

- Typically made up of eight bits

Character:

- Basic building block of information

Field:

- Name, number, or combination of characters that describes an aspect of a business object or activity

The Hierarchy of Data (continued)

Record:

- Collection of related data fields

File:

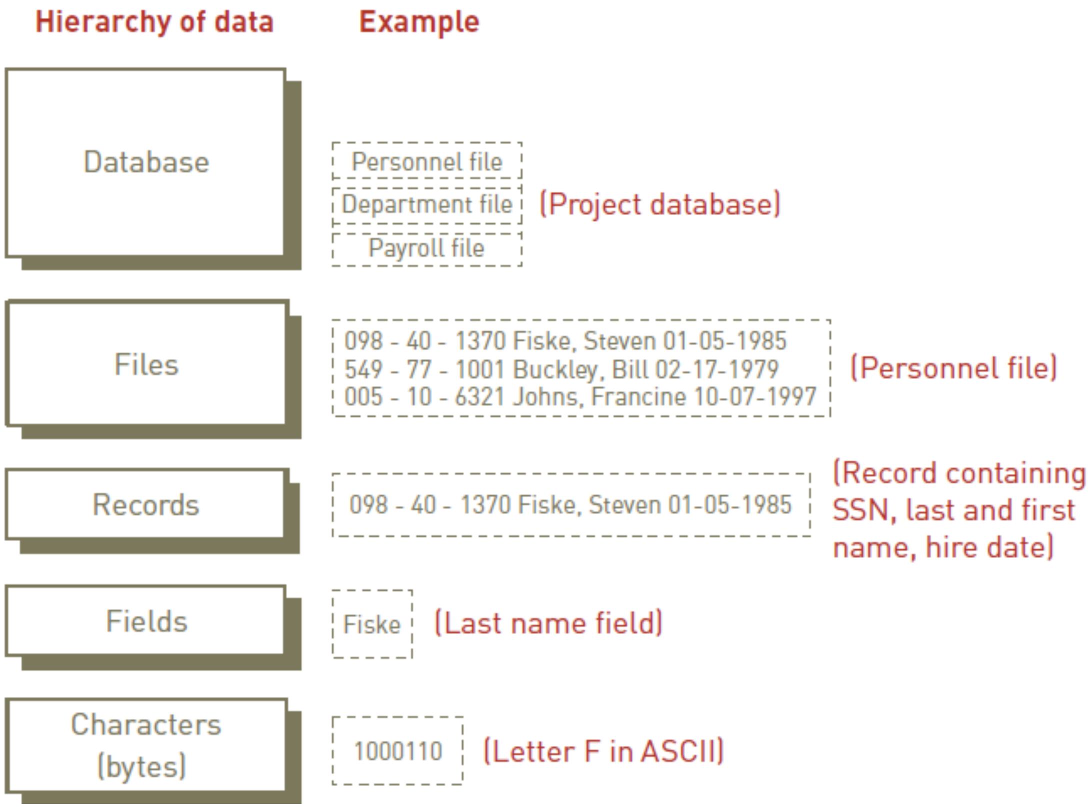
- Collection of related records

Database:

- Collection of integrated and related files

Hierarchy of data:

- Bits, characters, fields, records, files, and databases


Hierarchy of data
Example
Database
Personnel file
Department file
Payroll file
(Project database)
Files
098 - 40 - 1370 Fiske, Steven 01-05-1985
549 - 77 - 1001 Buckley, Bill 02-17-1979
005 - 10 - 6321 Johns, Francine 10-07-1997
(Personnel file)
Records
098 - 40 - 1370 Fiske, Steven 01-05-1985
(Record containing SSN, last and first name, hire date)
Fields
Fiske
(Last name field)
Characters (bytes)
1000110
(Letter F in ASCII)

Data Entities, Attributes, and Keys

Entity:

- General class of people, places, or things (objects) for which data is collected, stored, and maintained

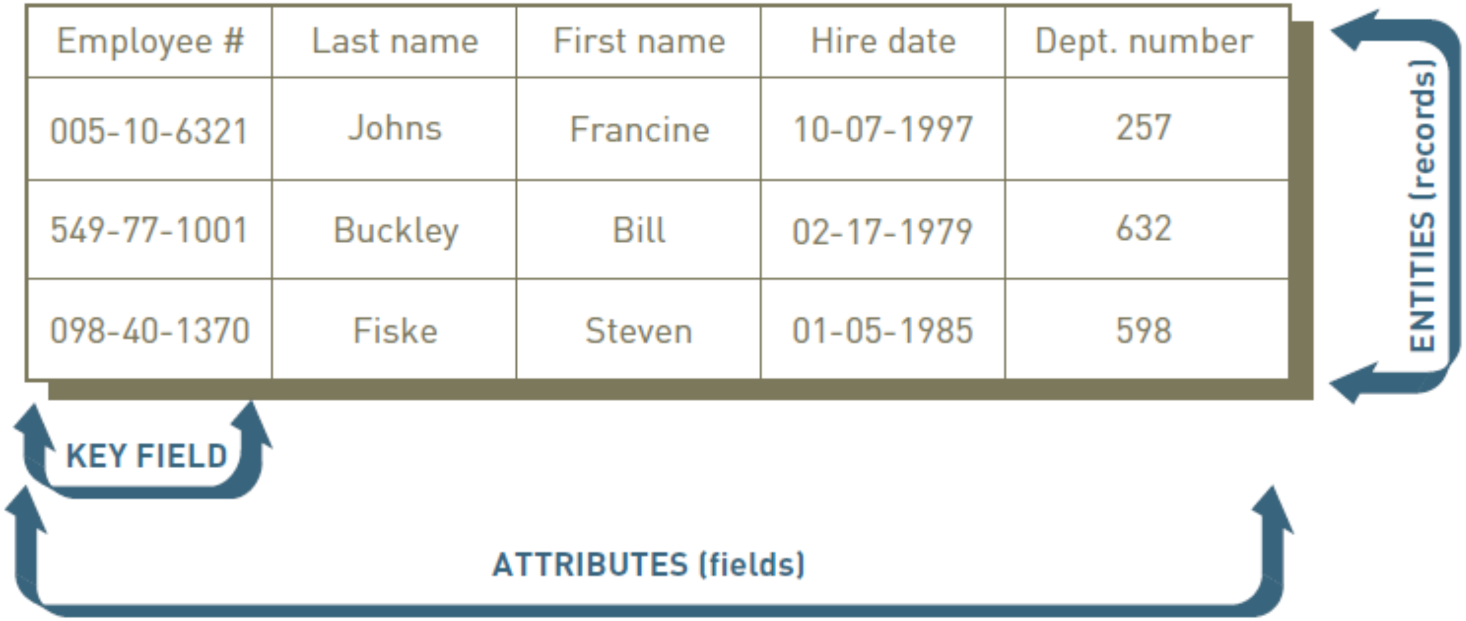
Attribute:

- Characteristic of an entity

Data item:

- Specific value of an attribute

Keys and Attributes



Employee #	Last name	First name	Hire date	Dept. number
005-10-6321	Johns	Francine	10-07-1997	257
549-77-1001	Buckley	Bill	02-17-1979	632
098-40-1370	Fiske	Steven	01-05-1985	598

ENTITIES (records)

KEY FIELD

ATTRIBUTES (fields)

Data Entities, Attributes, and Keys (continued)

Key:

- Field or set of fields in a record that is used to identify the record

Primary key:

- Field or set of fields that uniquely identifies the record

The Database Approach

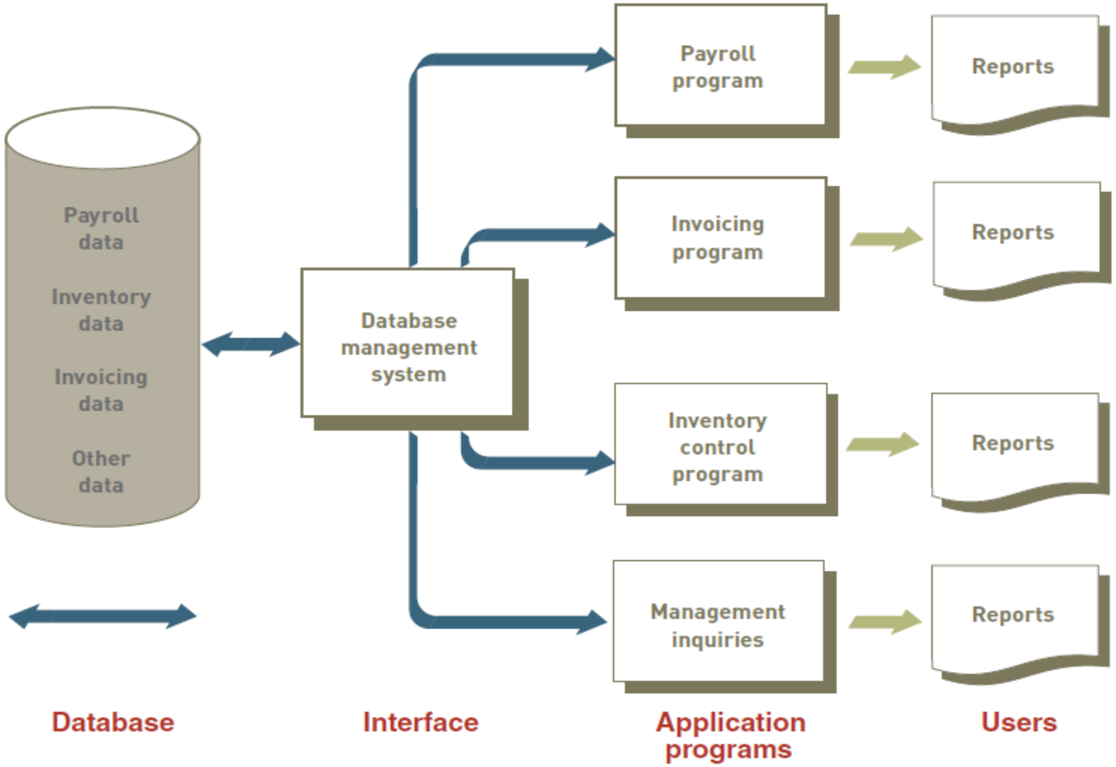
Traditional approach to data management:

- Each distinct operational system used data files dedicated to that system

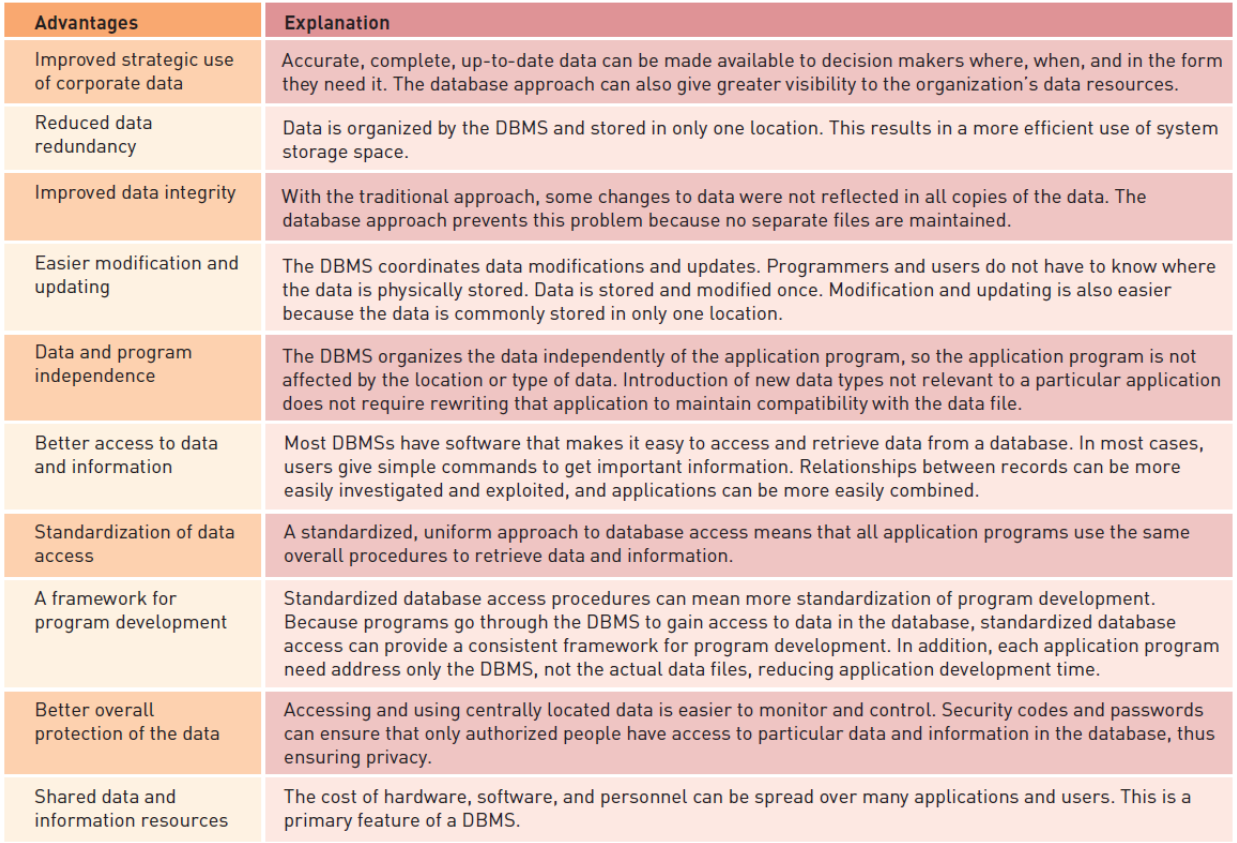
Database approach to data management:

- Pool of related data is shared by multiple application programs

The Database Approach to Data Management

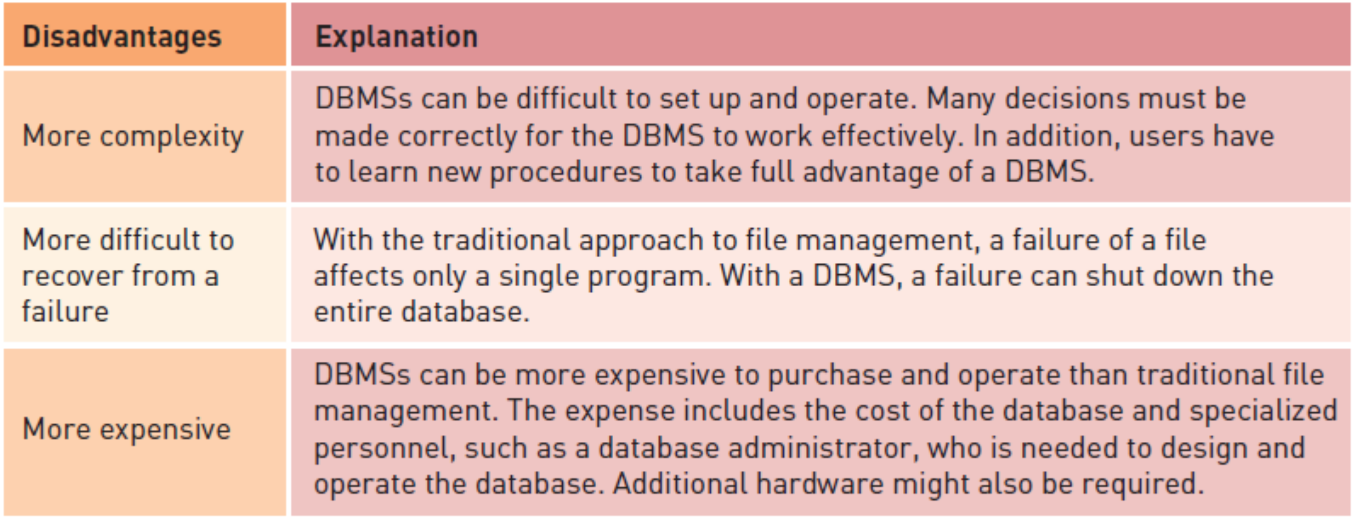


Advantages of the Database Approach



Advantages	Explanation
Improved strategic use of corporate data	Accurate, complete, up-to-date data can be made available to decision makers where, when, and in the form they need it. The database approach can also give greater visibility to the organization's data resources.
Reduced data redundancy	Data is organized by the DBMS and stored in only one location. This results in a more efficient use of system storage space.
Improved data integrity	With the traditional approach, some changes to data were not reflected in all copies of the data. The database approach prevents this problem because no separate files are maintained.
Easier modification and updating	The DBMS coordinates data modifications and updates. Programmers and users do not have to know where the data is physically stored. Data is stored and modified once. Modification and updating is also easier because the data is commonly stored in only one location.
Data and program independence	The DBMS organizes the data independently of the application program, so the application program is not affected by the location or type of data. Introduction of new data types not relevant to a particular application does not require rewriting that application to maintain compatibility with the data file.
Better access to data and information	Most DBMSs have software that makes it easy to access and retrieve data from a database. In most cases, users give simple commands to get important information. Relationships between records can be more easily investigated and exploited, and applications can be more easily combined.
Standardization of data access	A standardized, uniform approach to database access means that all application programs use the same overall procedures to retrieve data and information.
A framework for program development	Standardized database access procedures can mean more standardization of program development. Because programs go through the DBMS to gain access to data in the database, standardized database access can provide a consistent framework for program development. In addition, each application program need address only the DBMS, not the actual data files, reducing application development time.
Better overall protection of the data	Accessing and using centrally located data is easier to monitor and control. Security codes and passwords can ensure that only authorized people have access to particular data and information in the database, thus ensuring privacy.
Shared data and information resources	The cost of hardware, software, and personnel can be spread over many applications and users. This is a primary feature of a DBMS.

Disadvantages of the Database Approach



Disadvantages	Explanation
More complexity	DBMSs can be difficult to set up and operate. Many decisions must be made correctly for the DBMS to work effectively. In addition, users have to learn new procedures to take full advantage of a DBMS.
More difficult to recover from a failure	With the traditional approach to file management, a failure of a file affects only a single program. With a DBMS, a failure can shut down the entire database.
More expensive	DBMSs can be more expensive to purchase and operate than traditional file management. The expense includes the cost of the database and specialized personnel, such as a database administrator, who is needed to design and operate the database. Additional hardware might also be required.

Data Modeling and Database Characteristics

When building a database, an organization must consider:

- ***Content***: What data should be collected and at what cost?
- ***Access*:** What data should be provided to which users and when?
- ***Logical structure*:** How should data be arranged so that it makes sense to a given user?
- ***Physical organization*:** Where should data be physically located?

Data Modeling

Building a database requires two types of designs:

- **Logical design**:
  - Abstract model of how data should be structured and arranged to meet an organization’s information needs
- **Physical design**:
  - Starts from the logical database design and fine-tunes it for performance and cost considerations

Planned data redundancy:

- Done to improve system performance so that user reports or queries can be created more quickly

Data Modeling (continued)

Data model:

- Diagram of data entities and their relationships

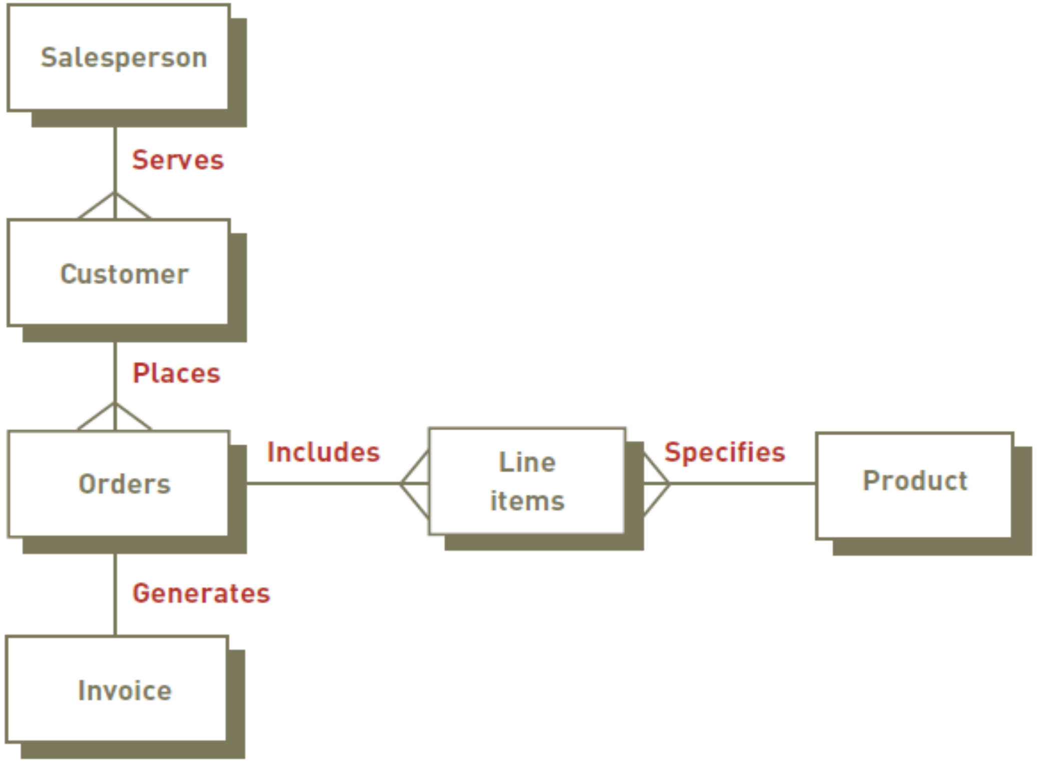
Enterprise data modeling:

- Starts by investigating the general data and information needs of the organization at the strategic level

Entity-relationship (ER) diagrams:

- Data models that use basic graphical symbols to show the organization of and relationships between data

An Entity-Relationship (ER) Diagram for a Customer Order Database



The Relational Database Model

Relational model:

- Describes data using a standard tabular format
- Each row of a table represents a data entity (record)
- Columns of the table represent attributes (fields)
- Domain: Allowable values for data attributes

Manipulating data

- **Selecting:**

  Eliminates rows according to certain criteria
- **Projecting:**

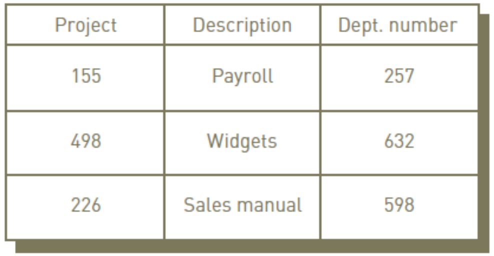
  Eliminates columns in a table
- **Joining:**

  Combines two or more tables
- **Linking:**

  Manipulating two or more tables that share at least one common data attribute

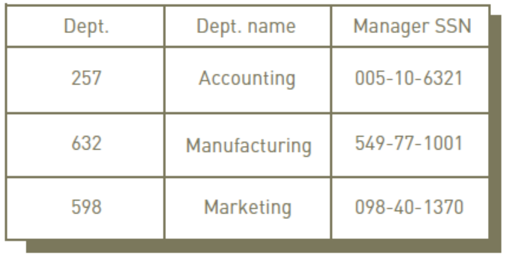
The Relational Database Model

Data Table 1: Project Table



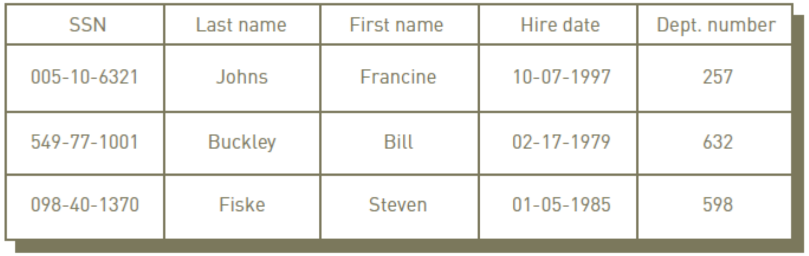
Project	Description	Dept. number
155	Payroll	257
498	Widgets	632
226	Sales manual	598

Data Table 2: Department Table



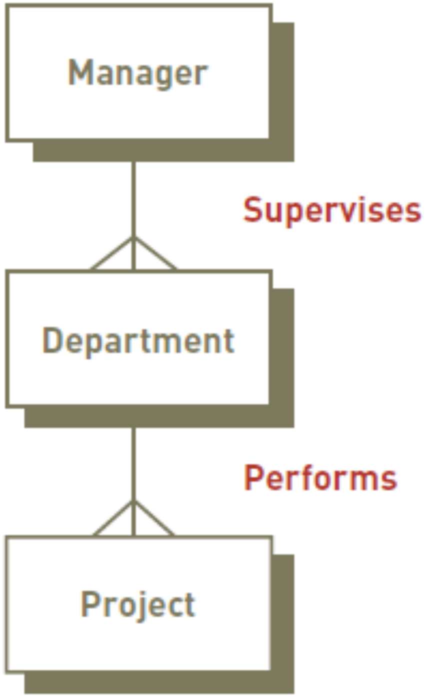
Dept.	Dept. name	Manager SSN
257	Accounting	005-10-6321
632	Manufacturing	549-77-1001
598	Marketing	098-40-1370

Data Table 3: Manager Table

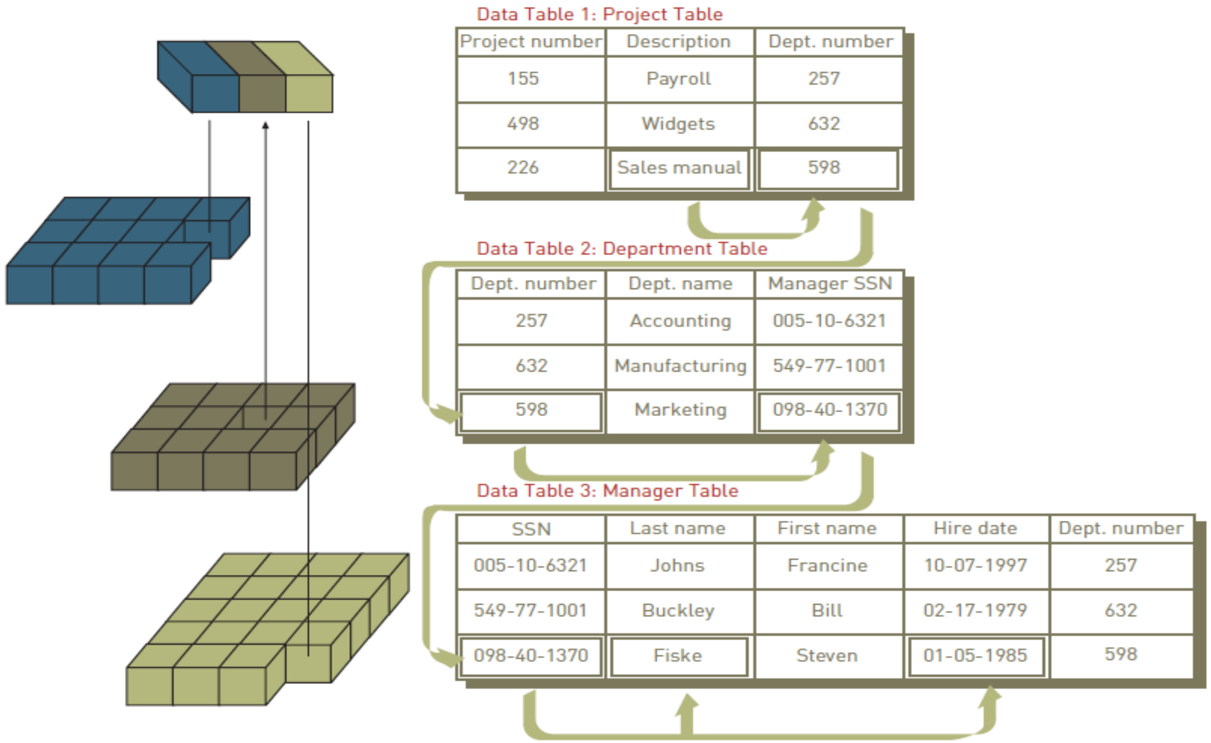


SSN	Last name	First name	Hire date	Dept. number
005-10-6321	Johns	Francine	10-07-1997	257
549-77-1001	Buckley	Bill	02-17-1979	632
098-40-1370	Fiske	Steven	01-05-1985	598

A Simplified ER Diagram Showing the Relationship Between the Manager, Department, and Project Tables



Linking Data Tables to Answer an Inquiry



Data Table 1: Project Table

Project number	Description	Dept. number
155	Payroll	257
498	Widgets	632
226	Sales manual	598

Data Table 2: Department Table

Dept. number	Dept. name	Manager SSN
257	Accounting	005-10-6321
632	Manufacturing	549-77-1001
598	Marketing	098-40-1370

Data Table 3: Manager Table

SSN	Last name	First name	Hire date	Dept. number
005-10-6321	Johns	Francine	10-07-1997	257
549-77-1001	Buckley	Bill	02-17-1979	632
098-40-1370	Fiske	Steven	01-05-1985	598

Database Management Systems

Creating and implementing the right database system:

- Ensures that the database will support both business activities and goals

Capabilities and types of database systems vary considerably

Overview of Database Types

Flat file:

- Simple database program whose records have no relationship to one another
- Example: OneNote

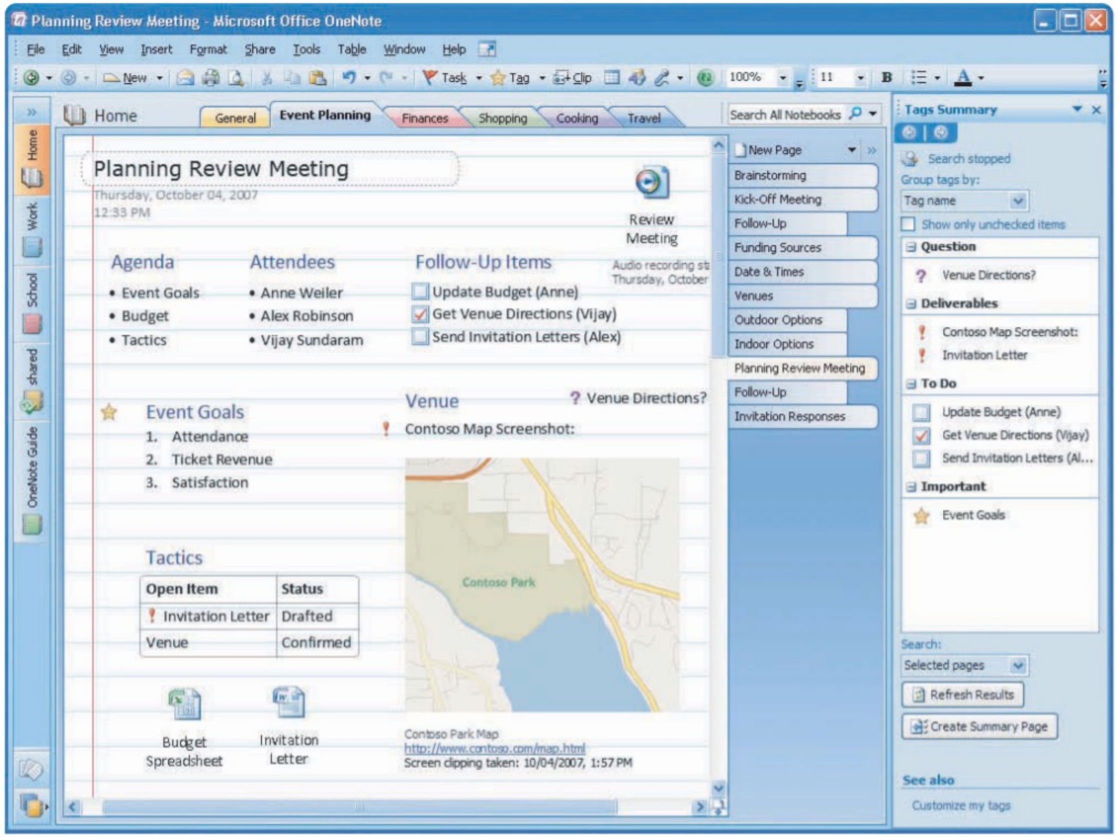
Single user:

- Only one person can use the database at a time
- Examples: Access, FileMaker Pro, and InfoPath

Multiple users:

- Allow dozens or hundreds of people to access the same database system at the same time
- Examples: Oracle, Sybase, and IBM

Microsoft OneNote



Planning Review Meeting

Thursday, October 04, 2007
12:33 PM

Agenda

- Event Goals
- Budget
- Tactics

Attendees

- Anne Weiler
- Alex Robinson
- Vijay Sundaram

Follow-Up Items

□ Update Budget (Anne)
☑ Get Venue Directions (Vijay)
□ Send Invitation Letters (Alex)

Event Goals

1. Attendance
2. Ticket Revenue
3. Satisfaction

Venue ? Venue Directions?

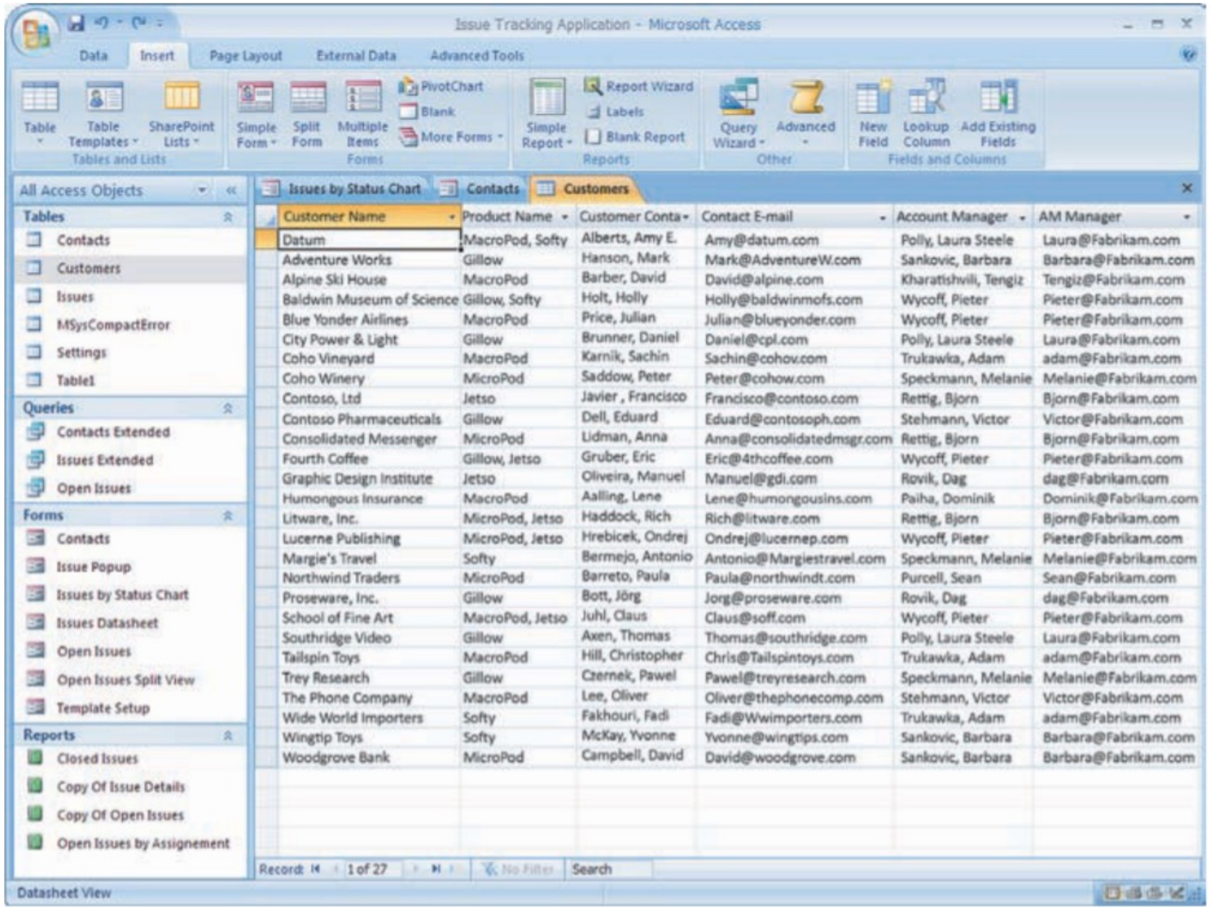
Contoso Map Screenshot:

Tactics

Open Item	Status
Invitation Letter	Drafted
Venue	Confirmed

Contoso Park Map
http://www.contoso.com/map.html
Screen clipping taken: 10/04/2007, 1:57 PM

Microsoft Office Access



Customer Name	Product Name	Customer Conta	Contact E-mail	Account Manager	AM Manager
Datum	MacroPod, Softy	Alberts, Amy E.	Amy@datum.com	Polly, Laura Steele	Laura@Fabrikam.com
Adventure Works	Gillow	Hanson, Mark	Mark@AdventureW.com	Sankovic, Barbara	Barbara@Fabrikam.com
Alpine Ski House	MacroPod	Barber, David	David@alpine.com	Kharatishvili, Tengiz	Tengiz@Fabrikam.com
Baldwin Museum of Science	Gillow, Softy	Holt, Holly	Holly@baldwinmofs.com	Wycoff, Pieter	Pieter@Fabrikam.com
Blue Yonder Airlines	MacroPod	Price, Julian	Julian@blueyonder.com	Wycoff, Pieter	Pieter@Fabrikam.com
City Power & Light	Gillow	Brunner, Daniel	Daniel@cpl.com	Polly, Laura Steele	Laura@Fabrikam.com
Coho Vineyard	MacroPod	Karnik, Sachin	Sachin@cohov.com	Trukawka, Adam	adam@Fabrikam.com
Coho Winery	MicroPod	Saddow, Peter	Peter@cohow.com	Speckmann, Melanie	Melanie@Fabrikam.com
Contoso, Ltd	Jetso	Javier , Francisco	Francisco@contoso.com	Rettig, Bjorn	Bjorn@Fabrikam.com
Contoso Pharmaceuticals	Gillow	Dell, Eduard	Eduard@contosoph.com	Stehmann, Victor	Victor@Fabrikam.com
Consolidated Messenger	MicroPod	Lidman, Anna	Anna@consolidatedmsgr.com	Rettig, Bjorn	Bjorn@Fabrikam.com
Fourth Coffee	Gillow, Jetso	Gruber, Eric	Eric@4thcoffee.com	Wycoff, Pieter	Pieter@Fabrikam.com
Graphic Design Institute	Jetso	Oliveira, Manuel	Manuel@gdi.com	Rovik, Dag	dag@Fabrikam.com
Humongous Insurance	MacroPod	Aalling, Lene	Lene@humongousins.com	Paiha, Dominik	Dominik@Fabrikam.com
Litware, Inc.	MicroPod, Jetso	Haddock, Rich	Rich@litware.com	Rettig, Bjorn	Bjorn@Fabrikam.com
Lucerne Publishing	MicroPod, Jetso	Hrebicek, Ondrej	Ondrej@lucernep.com	Wycoff, Pieter	Pieter@Fabrikam.com
Margie's Travel	Softy	Bermejo, Antonio	Antonio@Margiestravel.com	Speckmann, Melanie	Melanie@Fabrikam.com
Northwind Traders	MicroPod	Barreto, Paula	Paula@northwindt.com	Purcell, Sean	Sean@Fabrikam.com
Proseware, Inc.	Gillow	Bott, Jörg	Jorg@proseware.com	Rovik, Dag	dag@Fabrikam.com
School of Fine Art	MacroPod, Jetso	Juhl, Claus	Claus@soff.com	Wycoff, Pieter	Pieter@Fabrikam.com
Southridge Video	Gillow	Axen, Thomas	Thomas@southridge.com	Polly, Laura Steele	Laura@Fabrikam.com
Tailspin Toys	MacroPod	Hill, Christopher	Chris@Tailspintoys.com	Trukawka, Adam	adam@Fabrikam.com
Trey Research	Gillow	Czernek, Pawel	Pawel@treyresearch.com	Speckmann, Melanie	Melanie@Fabrikam.com
The Phone Company	MacroPod	Lee, Oliver	Oliver@thephonecomp.com	Stehmann, Victor	Victor@Fabrikam.com
Wide World Importers	Softy	Fakhouri, Fadi	Fadi@Wwimporters.com	Trukawka, Adam	adam@Fabrikam.com
Wingtip Toys	Softy	McKay, Yvonne	Yvonne@wingtips.com	Sankovic, Barbara	Barbara@Fabrikam.com
Woodgrove Bank	MicroPod	Campbell, David	David@woodgrove.com	Sankovic, Barbara	Barbara@Fabrikam.com

Providing a User View

Schema:

- Used to describe the entire database
- Can be part of the database or a separate schema file

DBMS:

- Can reference a schema to find where to access the requested data in relation to another piece of data

Creating and Modifying the Database

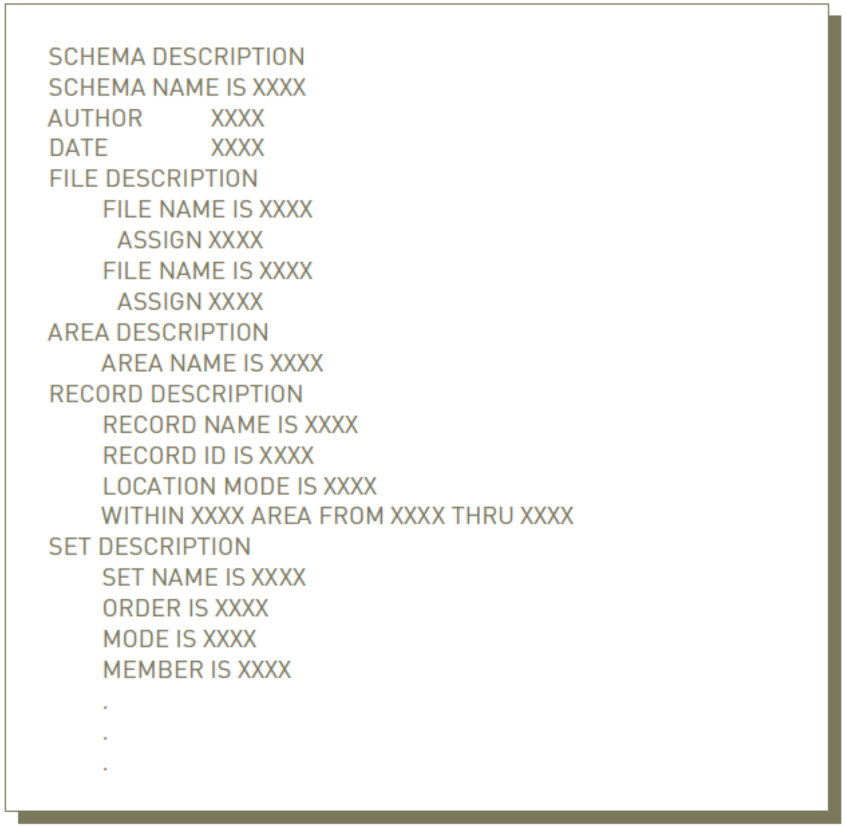
Data definition language (DDL):

- Collection of instructions and commands used to define and describe data and relationships in a specific database
- Allows database's creator to describe data and relationships that are to be contained in the schema

Data dictionary:

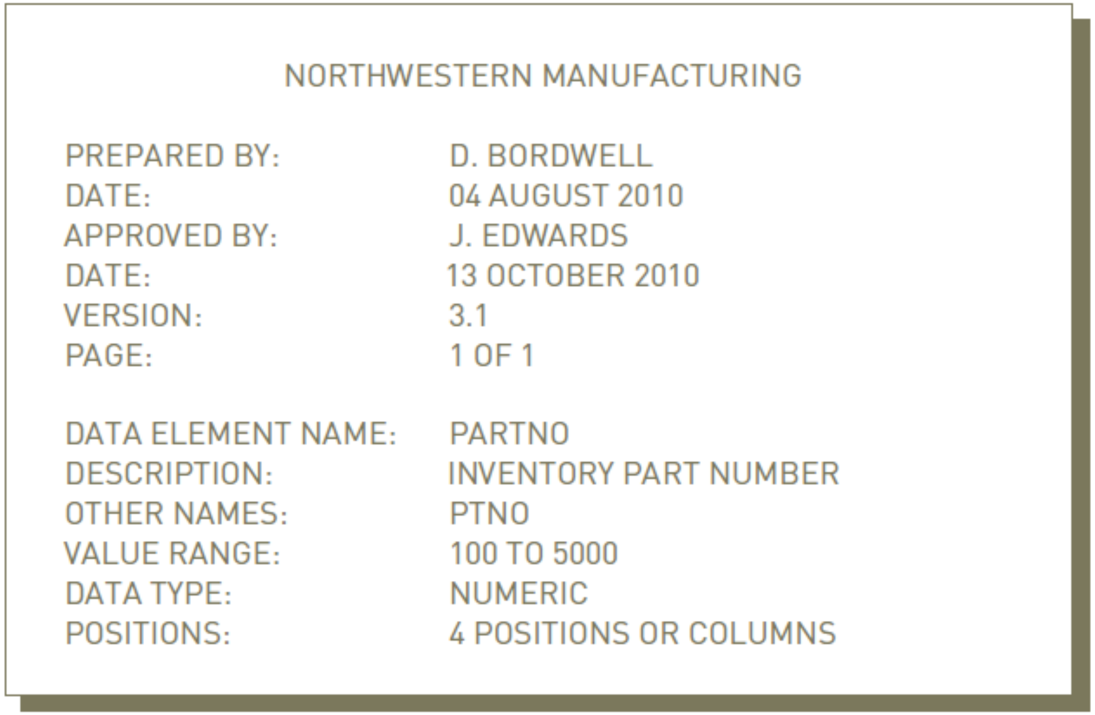
- Detailed description of all the data used in the database
- The data dictionary contains the following data:
    - Name of the data item
    - Aliases or other names that may be used to describe the item
    - Range of values that can be used
    - Type of data (such as alphanumeric or numeric)
    - Amount of storage needed for the item
    - Notation of the person responsible for updating it and the various users who can access it
    - List of reports that use the data item

Using a Data Definition Language to Define a Schema



```
SCHEMA DESCRIPTION
SCHEMA NAME IS XXXX
AUTHOR          XXXX
DATE            XXXX
FILE DESCRIPTION
     FILE NAME IS XXXX
       ASSIGN XXXX
     FILE NAME IS XXXX
       ASSIGN XXXX
AREA DESCRIPTION
     AREA NAME IS XXXX
RECORD DESCRIPTION
     RECORD NAME IS XXXX
     RECORD ID IS XXXX
     LOCATION MODE IS XXXX
     WITHIN XXXX AREA FROM XXXX THRU XXXX
SET DESCRIPTION
     SET NAME IS XXXX
     ORDER IS XXXX
     MODE IS XXXX
     MEMBER IS XXXX
     .
     .
     .
```

A Typical Data Dictionary Entry



NORTHWESTERN MANUFACTURING

PREPARED BY:	D. BORDWELL
DATE:	04 AUGUST 2010
APPROVED BY:	J. EDWARDS
DATE:	13 OCTOBER 2010
VERSION:	3.1
PAGE:	1 OF 1
DATA ELEMENT NAME:	PARTNO
DESCRIPTION:	INVENTORY PART NUMBER
OTHER NAMES:	PTNO
VALUE RANGE:	100 TO 5000
DATA TYPE:	NUMERIC
POSITIONS:	4 POSITIONS OR COLUMNS

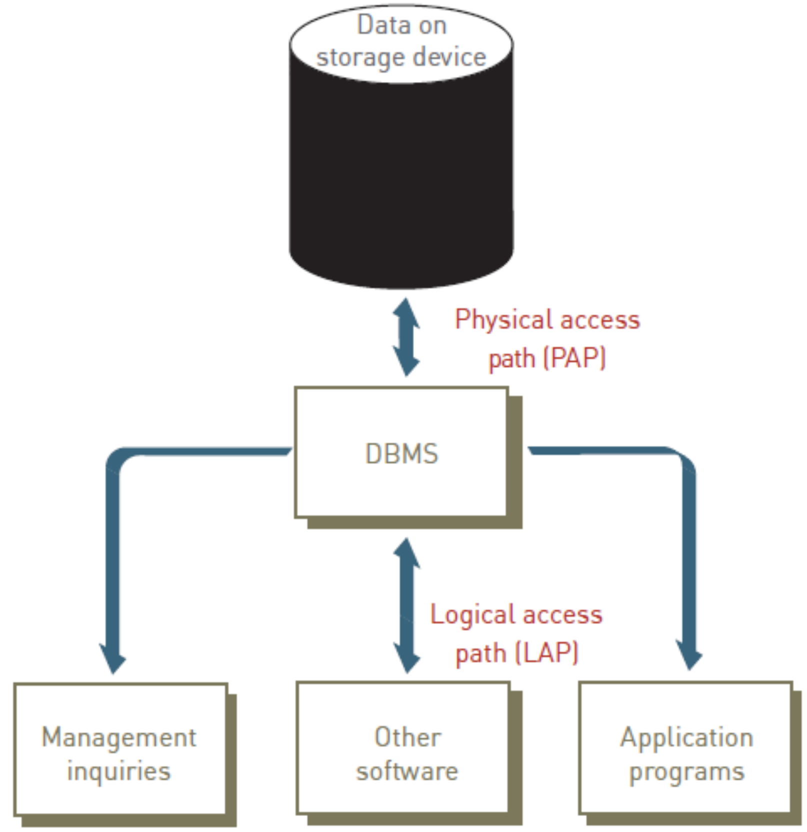
Storing and Retrieving Data

When an application program needs data:

- It requests the data through the DBMS

- Concurrency control:
  - Method of dealing with a situation in which two or more users or applications need to access the same record at the same time

Logical and Physical Access Paths



Manipulating Data and Generating Reports

Data manipulation language (**DML**):

- Commands that manipulate the data in a database

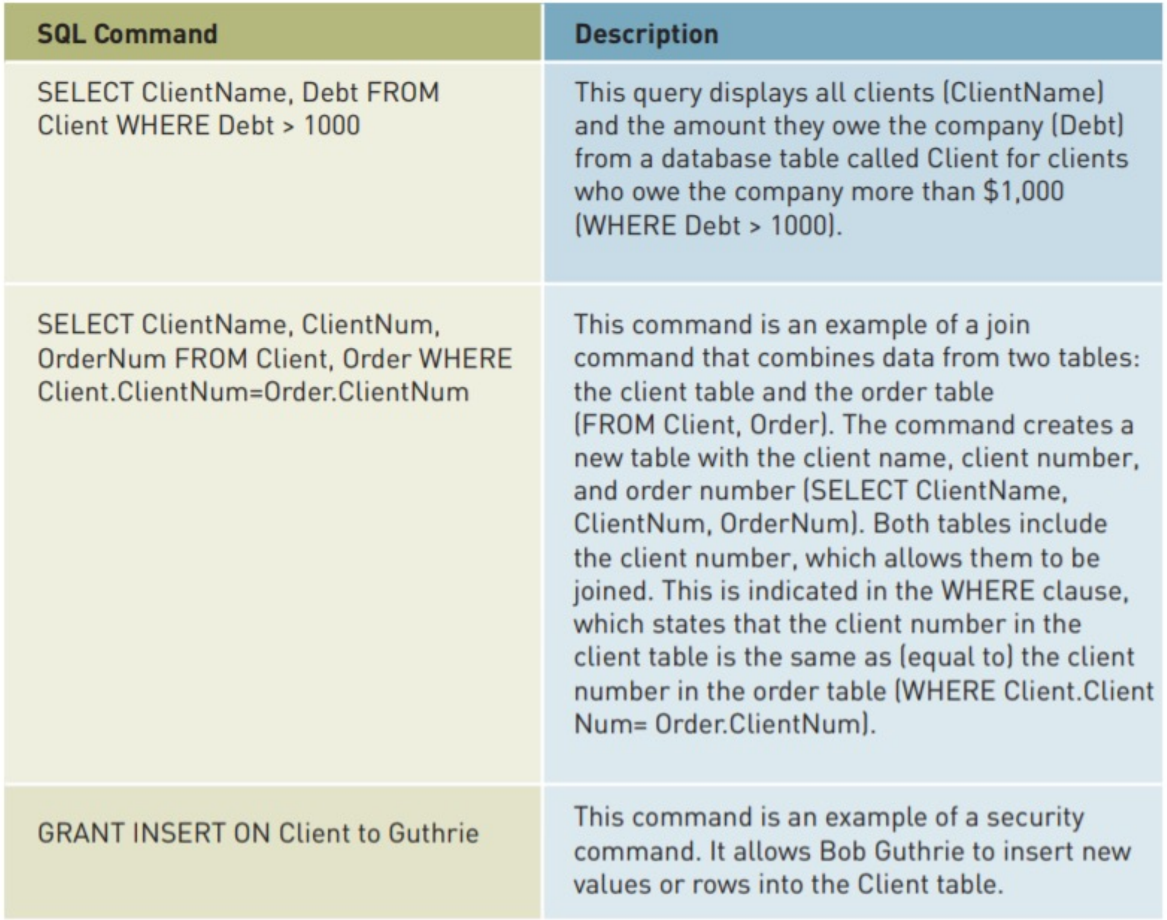
Structured Query Language (**SQL**):

- Adopted by the American National Standards Institute (ANSI) as the standard query language for relational databases
- Example for query: SELECT * FROM EMPLOYEE WHERE JOB_CLASSIFICATION = “C2”

Once a database has been set up and loaded with data:

- It can produce reports, documents, and other outputs

Examples of SQL Commands



SQL Command	Description
SELECT ClientName, Debt FROM Client WHERE Debt > 1000	This query displays all clients (ClientName) and the amount they owe the company (Debt) from a database table called Client for clients who owe the company more than $1,000 (WHERE Debt > 1000).
SELECT ClientName, ClientNum, OrderNum FROM Client, Order WHERE Client.ClientNum=Order.ClientNum	This command is an example of a join command that combines data from two tables: the client table and the order table (FROM Client, Order). The command creates a new table with the client name, client number, and order number (SELECT ClientName, ClientNum, OrderNum). Both tables include the client number, which allows them to be joined. This is indicated in the WHERE clause, which states that the client number in the client table is the same as (equal to) the client number in the order table (WHERE Client.Client Num= Order.ClientNum).
GRANT INSERT ON Client to Guthrie	This command is an example of a security command. It allows Bob Guthrie to insert new values or rows into the Client table.

Database and Data Administration

Database Administrator (DBA):

- Works with users to decide the content of the database
- Works with programmers as they build applications to ensure that their programs comply with database management system standards and conventions

Data administrator:

- Responsible for defining and implementing consistent principles for a variety of data issues

Popular Database Management Systems

Popular DBMSs for end users:

- Microsoft's Access and FileMaker Pro

Database as a Service (DaaS):

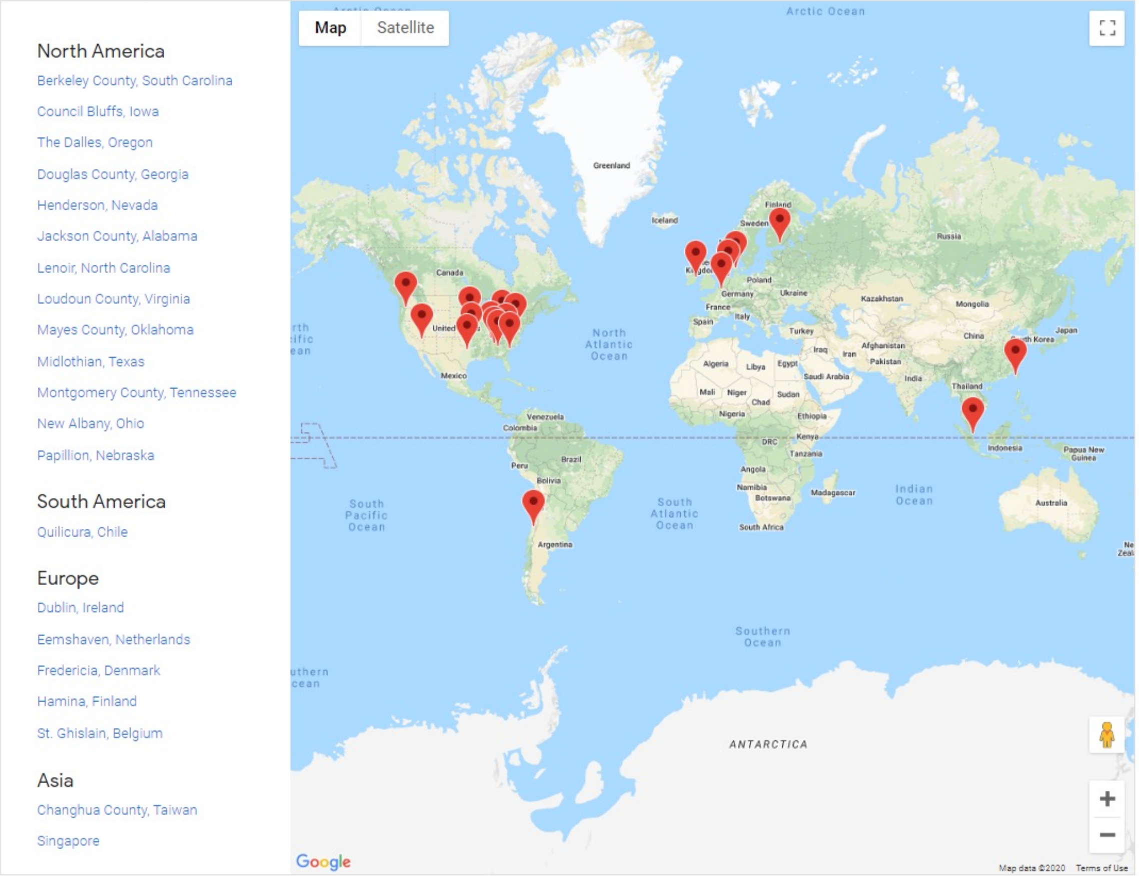
- Emerging database system
- Database administration is provided by the service provider
- The database is stored on a service provider's servers and accessed by the client over a network

Selecting a Database Management System

Important characteristics of databases to consider:

- Database size. The number of records or files in the database
- Database cost. The purchase or lease costs of the database
- Concurrent users. The number of people who need to use the database at the same time (the number of concurrent users)
- Performance. How fast the database is able to update records
- Integration. The ability to be integrated with other applications and databases
- Vendor. The reputation and financial stability of the database vendor

Google Data Center Locations



North America

Berkeley County, South Carolina

Council Bluffs, Iowa

The Dalles, Oregon

Douglas County, Georgia

Henderson, Nevada

Jackson County, Alabama

Lenoir, North Carolina

Loudoun County, Virginia

Mayes County, Oklahoma

Midlothian, Texas

Montgomery County, Tennessee

New Albany, Ohio

Papillion, Nebraska

South America

Quilicura, Chile

Europe

Dublin, Ireland

Eemshaven, Netherlands

Fredericia, Denmark

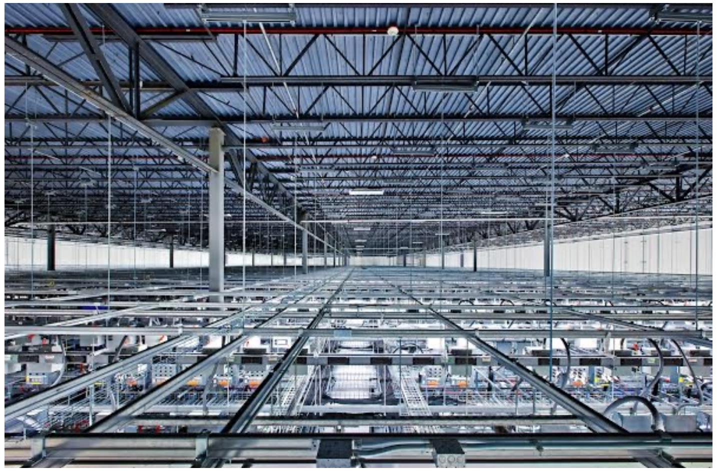
Hamina, Finland

St. Ghislain, Belgium

Asia

Changhua County, Taiwan

Singapore



Using Databases with Other Software

DBMSs can act as front-end or back-end applications:

- **Front-end applications** interact directly with people
- **Back-end applications** interact with other programs or applications

Database Applications

Today’s database applications manipulate the content of a database to produce useful information

Common manipulations:

- Searching, filtering, synthesizing, and assimilating data contained in a database using a number of database applications

Linking Databases to the Internet

Semantic Web:

- Developing a seamless integration of traditional databases with the Internet
- Provides metadata with all Web content using technology called the Resource Description Framework (RDF)

Data Warehouses, Data Marts, and Data Mining

Data warehouse:

- Database that holds business information from many sources in the enterprise

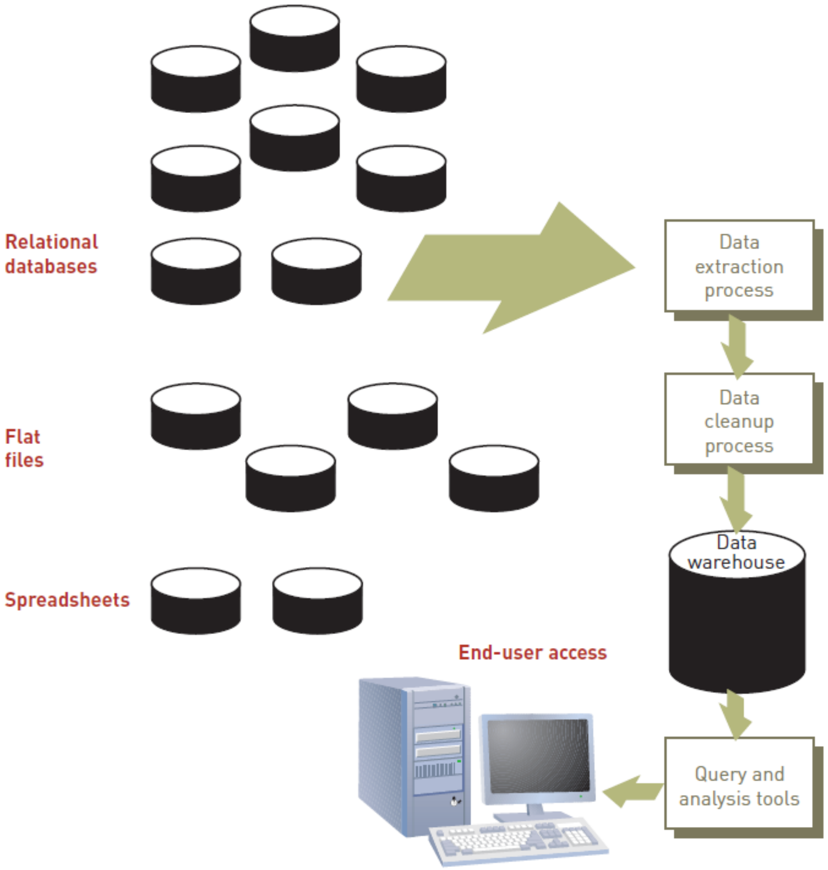
Data mart:

- Subset of a data warehouse

Data mining:

- Information-analysis tool that involves the automated discovery of patterns and relationships in a data warehouse

Elements of a Data Warehouse

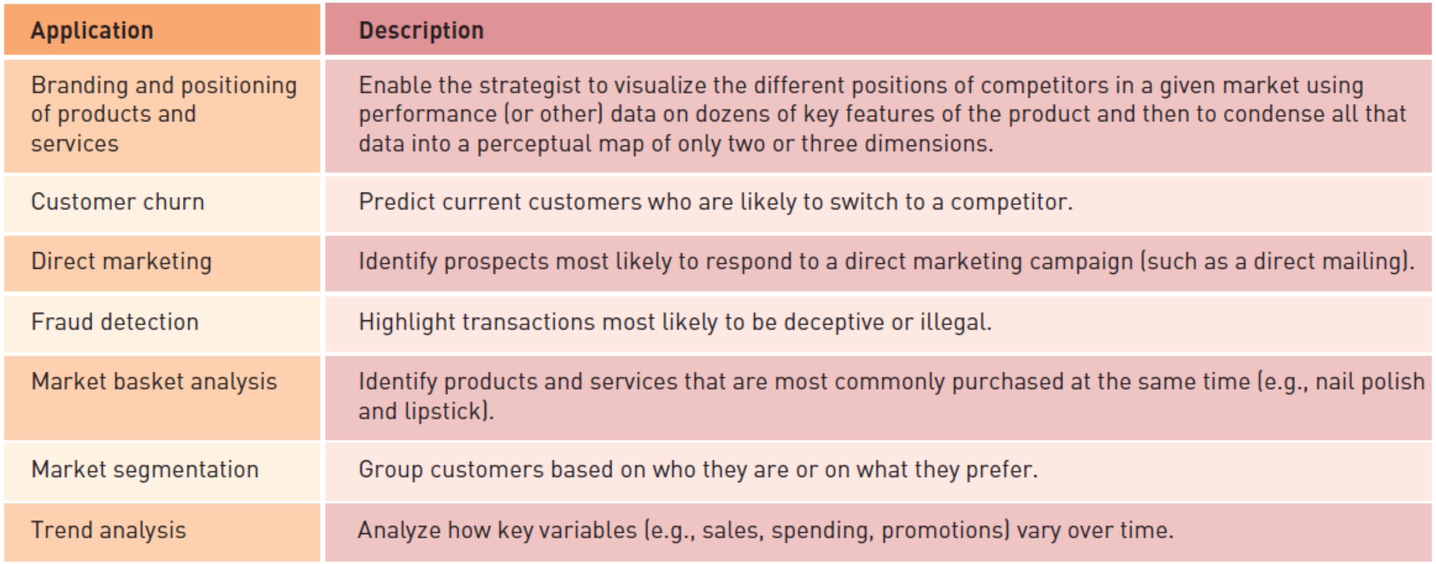


Data Warehouses, Data Marts, and Data Mining (continued)

Predictive analysis:

- Form of data mining that combines historical data with assumptions about future conditions to predict outcomes of events
- Used by retailers to upgrade occasional customers into frequent purchasers
- Software can be used to analyze a company's customer list and a year's worth of sales data to find new market segments

Common Data-Mining Applications



Application	Description
Branding and positioning of products and services	Enable the strategist to visualize the different positions of competitors in a given market using performance (or other) data on dozens of key features of the product and then to condense all that data into a perceptual map of only two or three dimensions.
Customer churn	Predict current customers who are likely to switch to a competitor.
Direct marketing	Identify prospects most likely to respond to a direct marketing campaign (such as a direct mailing).
Fraud detection	Highlight transactions most likely to be deceptive or illegal.
Market basket analysis	Identify products and services that are most commonly purchased at the same time (e.g., nail polish and lipstick).
Market segmentation	Group customers based on who they are or on what they prefer.
Trend analysis	Analyze how key variables (e.g., sales, spending, promotions) vary over time.

Business Intelligence

Involves gathering enough of the right information:

- In a timely manner and usable form and analyzing it to have a positive impact on business strategy, tactics, or operations

Competitive intelligence:

- Limited to information about competitors and the ways that knowledge affects strategy, tactics, and operations

Counterintelligence:

- Steps organization takes to protect information sought by “hostile” intelligence gatherers

Data loss prevention (DLP):

- Refers to systems designed to lock down data within an organization
- Powerful tool for counterintelligence
- A necessity in complying with government regulations that require companies to safeguard private customer data

Distributed Databases

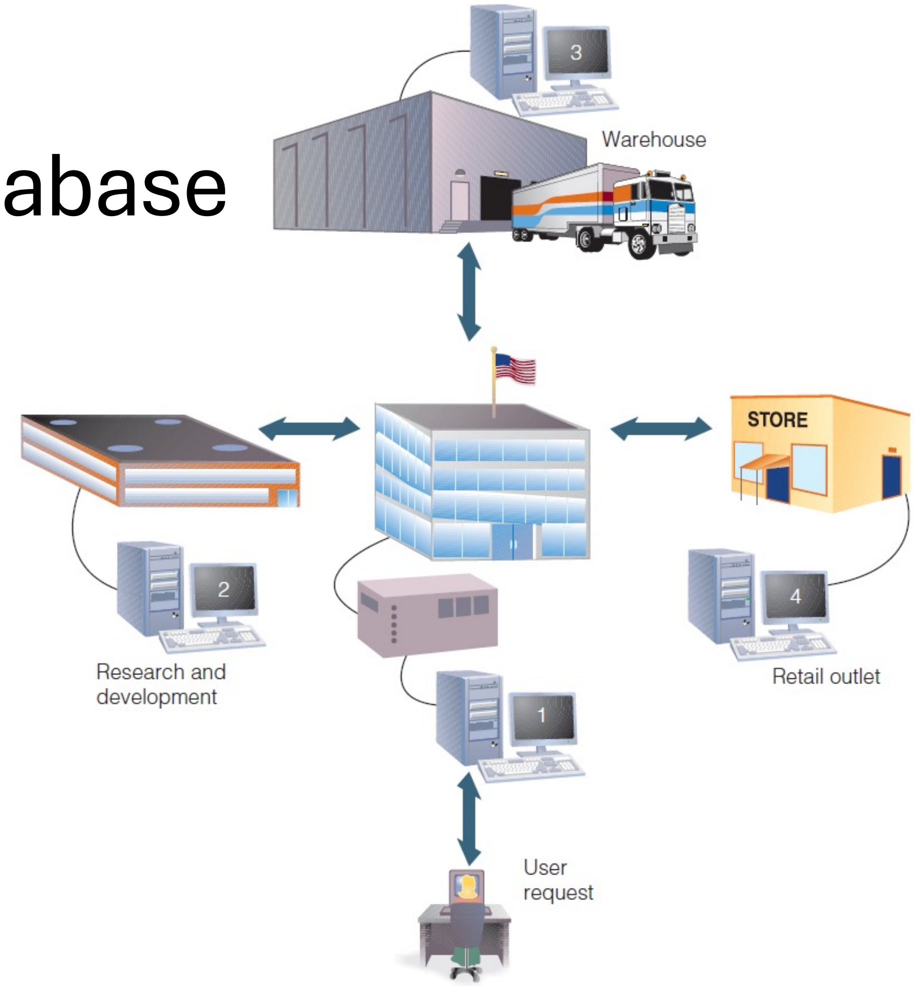
Distributed database:

- Database in which the data may be spread across several smaller databases connected via telecommunications devices
- Gives corporations more flexibility in how databases are organized and used

Replicated database:

- Holds a duplicate set of frequently used data

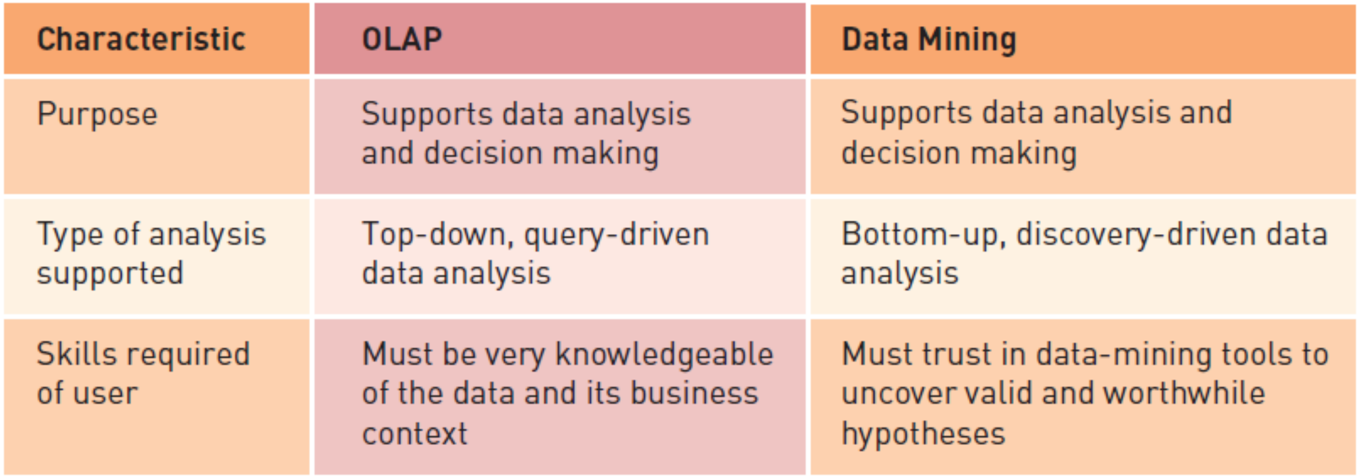
The Use of a Distributed Database



Online Analytical Processing (OLAP)

- Software that allows users to explore data from a number of different perspectives
- Provides top-down, query-driven data analysis
- Requires repetitive testing of user-originated theories
- Requires a great deal of human ingenuity and interaction with the database to find information

Comparison of OLAP and Data Mining



Characteristic	OLAP	Data Mining
Purpose	Supports data analysis and decision making	Supports data analysis and decision making
Type of analysis supported	Top-down, query-driven data analysis	Bottom-up, discovery-driven data analysis
Skills required of user	Must be very knowledgeable of the data and its business context	Must trust in data-mining tools to uncover valid and worthwhile hypotheses

Object-Relational Database Management Systems

Object-oriented database:

- Stores both data and its processing instructions
- Uses an object-oriented database management system (OODBMS) to provide a user interface and connections to other programs

Object-relational database management system (ORDBMS):

- Provides the ability for third parties to add new data types and operations to the database

Visual, Audio, and Other Database Systems

Visual databases:

- Can be stored in some object-relational databases or special-purpose database systems

Virtual database systems:

- Allow different databases to work together as a unified database system

Spatial data technology:

- Using database to store and access data according to the locations it describes

Information and Decision Support Systems

Decision Making and Problem Solving

In most cases, strategic planning and overall goals of the organization set the course for decision making

Information systems:

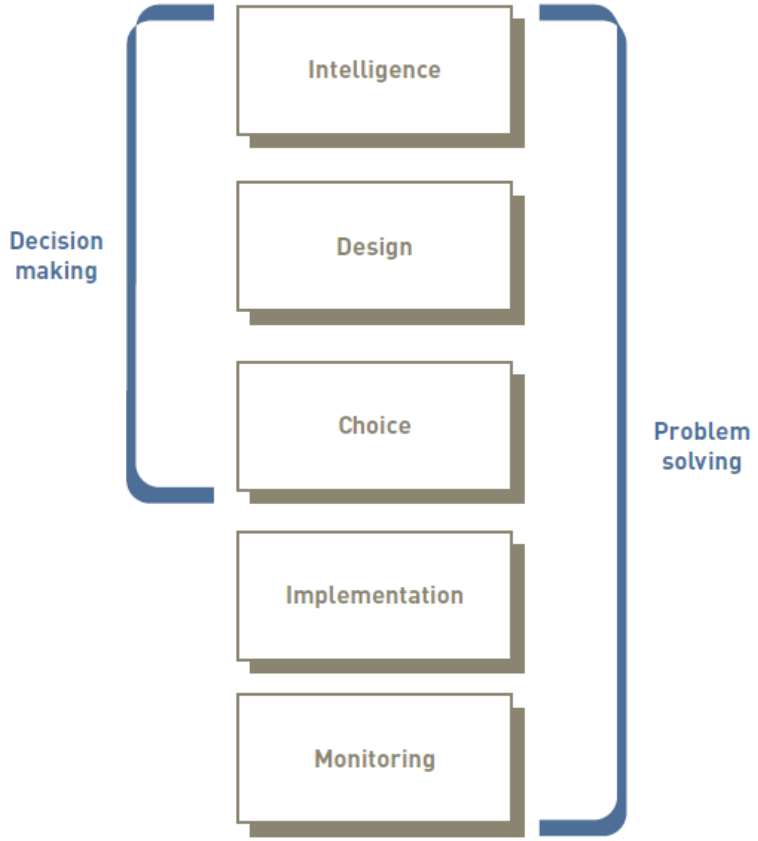
- Assist with problem solving, helping people make better decisions and save lives

Decision Making as a Component of Problem Solving

Decision-making phase:

- Intelligence stage:
  - Identify and define potential problems or opportunities
- Design stage:
  - Develop alternative solutions to the problem and evaluate their feasibility
- Choice stage:
  - Select a course of action
- Problem solving:
  - Includes and goes beyond decision making
  - Includes implementation stage
- Monitoring stage:
  - Decision makers evaluate the implementation

How Decision Making Relates to Problem Solving



Programmed versus Nonprogrammed Decisions

- Programmed decision:
  - Made using a rule, procedure, or quantitative method
  - Easy to computerize using traditional information systems
- Nonprogrammed decision:
  - Decision that deals with unusual or exceptional situations
  - Not easily quantifiable

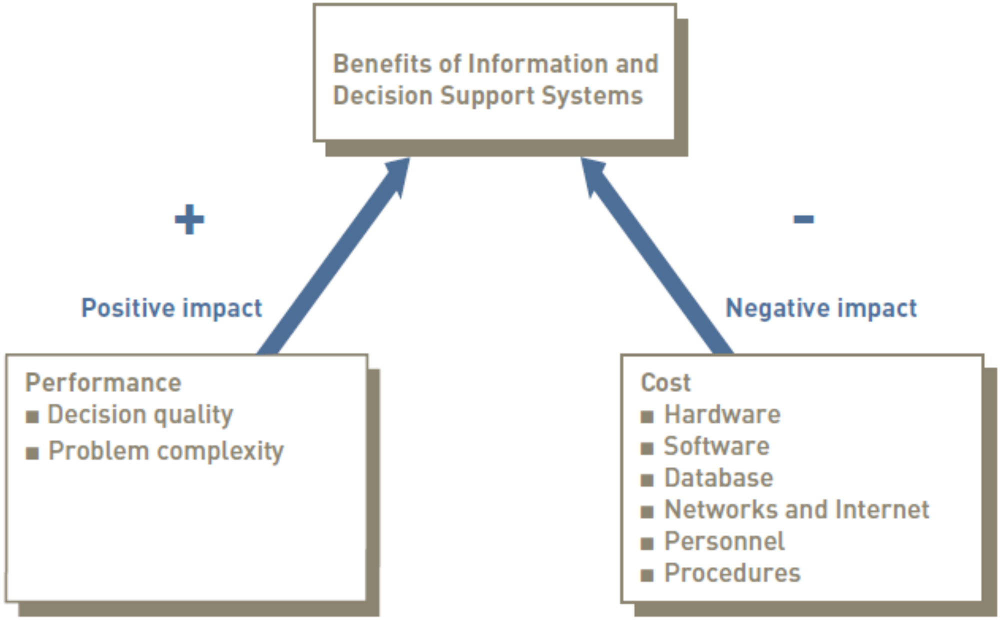
Optimization, Satisficing, and Heuristic Approaches

- Optimization model:
  - Finds the best solution, usually the one that will best help the organization meet its goals
- Satisficing model:
  - Finds a good, but not necessarily the best, problem solution
- Heuristics:
  - Commonly accepted guidelines or procedures that usually find a good solution

The Benefits of Information and Decision Support Systems

- Decision support systems:
  - Performance is typically a function of decision quality and problem complexity
- Problem complexity:
  - Depends on how hard the problem is to solve and implement

The Benefits of Information and Decision Support Systems (continued)



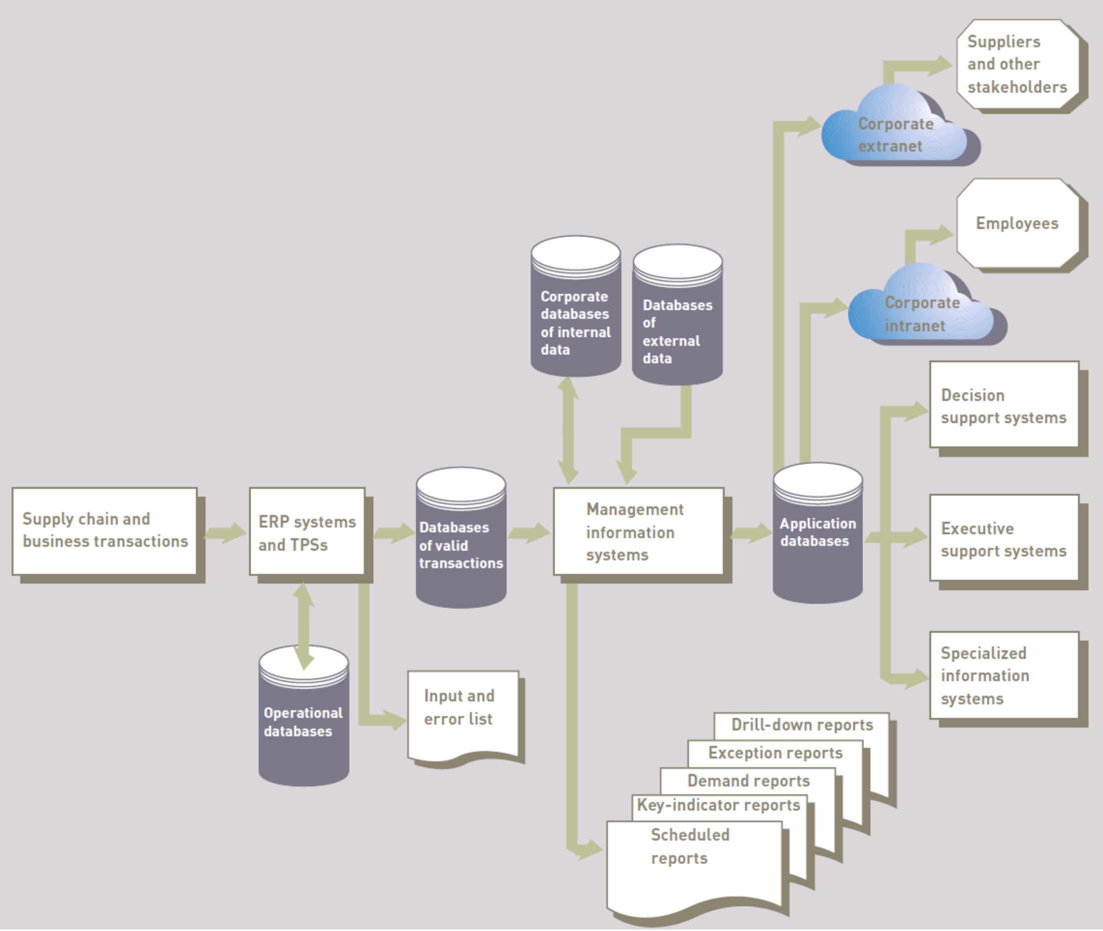
An Overview of Management Information Systems

- Management information system (MIS)
  - Integrated collection of people, procedures, databases, and devices
  - Can give the organization a competitive advantage

Management Information Systems in Perspective

- Purpose of an MIS:
  - To help an organization achieve its goals
  - Provide the right information to the right person in the right format at the right time
- Business transactions:
  - Can enter the organization through traditional methods, or via the Internet, or via an extranet

Sources of Managerial Information


Supply chain and business transactions
ERP systems and TPSs
Operational databases
Databases of valid transactions
Input and error list
Corporate databases of internal data
Databases of external data
Management information systems
Drill-down reports
Exception reports
Demand reports
Key-indicator reports
Scheduled reports
Application databases
Corporate extranet
Suppliers and other stakeholders
Corporate intranet
Employees
Decision support systems
Executive support systems
Specialized information systems

Inputs to a Management Information System

- Internal data sources:
  - TPS and ERP systems and related databases
- External data sources:
  - Customers, suppliers, competitors, and stockholders whose data is not already captured by the TPS and ERP systems
  - Business intelligence:
    - Can be used to turn a database into useful information throughout the organization

Outputs of a Management Information System

- Scheduled reports:
    - Produced periodically, such as daily, weekly, or monthly
    - *Key-indicator* report summarizes the previous day's critical activities
- Demand reports:
    - Developed to provide certain information upon request

Outputs of a Management Information System (continued)

- Exception reports:
  - Automatically produced when a situation is unusual or requires management action
  - Trigger points should be set carefully
- Drill-down reports:
  - Provide increasingly detailed data about a situation

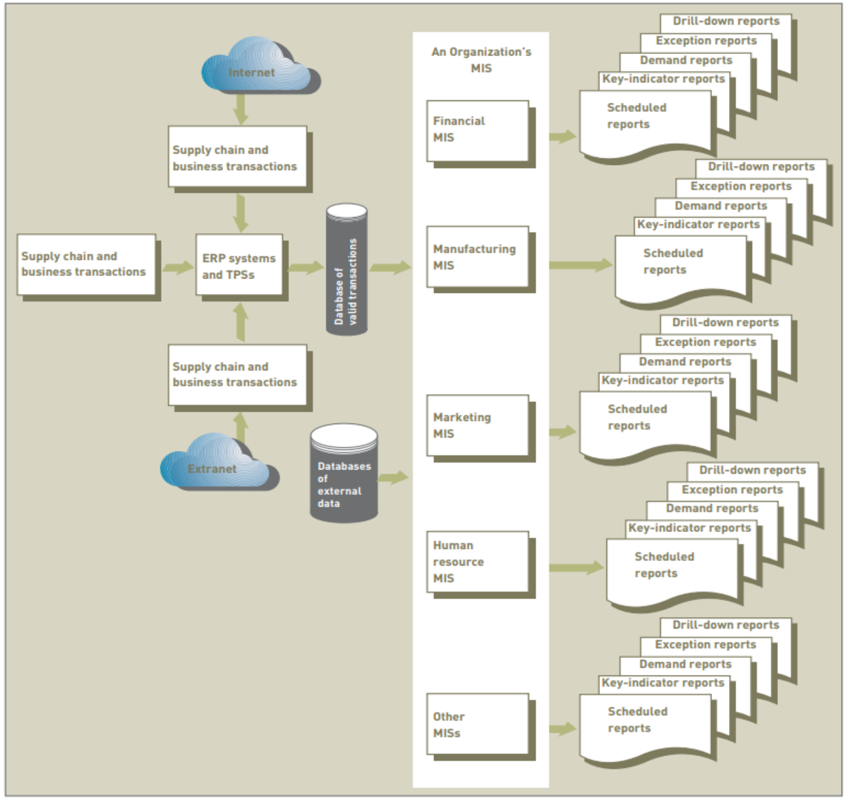
Characteristics of a Management Information System

- MISs perform the following functions:
  - Provide reports with fixed and standard formats
  - Produce hard-copy and soft-copy reports
  - Use internal data stored in computer system
  - Allow users to develop custom reports
  - Require user requests for reports developed by systems personnel

Functional Aspects of the MIS

- Most organizations are structured along functional lines or areas
- MIS can be divided along functional lines to produce reports tailored to individual functions

An Organization’s MIS



Financial Management Information Systems

Financial MIS:

- Provides financial information to executives and others

Functions:

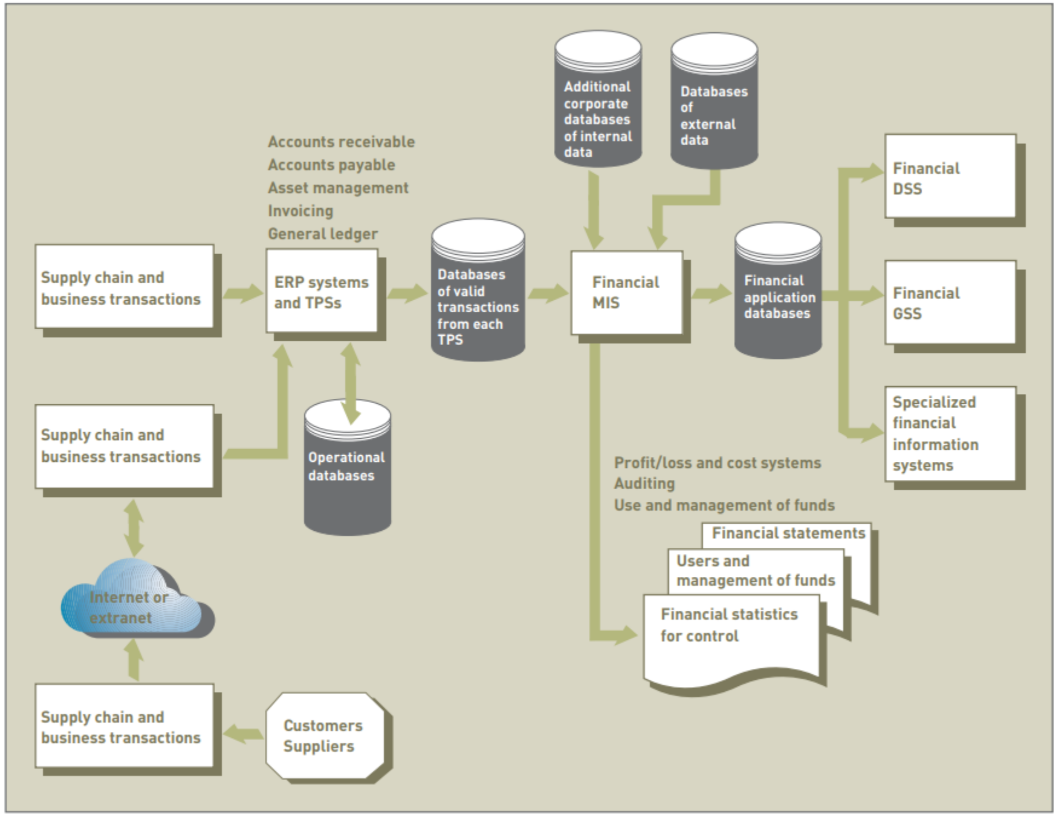
- Integrate financial and operational information from multiple sources, including the Internet, into a single system
- Provide easy access to data for both financial and nonfinancial users, often through the use of a corporate intranet to access corporate Web pages of financial data and information
- Make financial data immediately available to shorten analysis turnaround time
- Enable analysis of financial data along multiple dimensions—time, geography, product, plant, and customer
- Analyze historical and current financial activity
- Monitor and control the use of funds over time

Financial Management Information Systems

Some of the financial MIS subsystems and outputs:

- **profit center**: A department within an organization that focuses on generating profits.
  - revenue center: A division within a company that generates sales or revenues.
  - cost center A division within a company that does not directly generate revenue.
- **auditing**: Analyzing the financial condition of an organization and determining whether financial statements and reports produced by the financial MIS are accurate.
  - internal auditing: Auditing performed by individuals within the organization.
  - external auditing: Auditing performed by an outside group.
- **uses and management of funds**

Overview of a Financial MIS


Supply chain and business transactions
Supply chain and business transactions
Supply chain and business transactions
Internet or extranet
Customers
Suppliers
Accounts receivable
Accounts payable
Asset management
Invoicing
General ledger
ERP systems and TPSs
Operational databases
Databases of valid transactions from each TPS
Additional corporate databases of internal data
Databases of external data
Financial MIS
Financial application databases
Financial DSS
Financial GSS
Specialized financial information systems
Profit/loss and cost systems
Auditing
Use and management of funds
Financial statements
Users and management of funds
Financial statistics for control

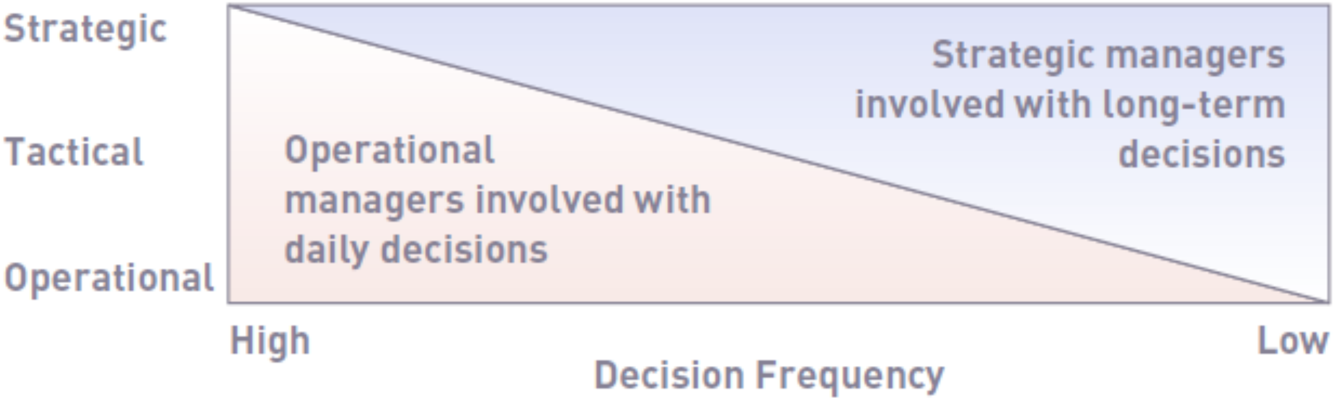
An Overview of Decision Support Systems

- DSS:
  - Organized collection of people, procedures, software, databases, and devices used to help make decisions that solve problems
  - Used at all levels
- Focus of a DSS:
  - Is on decision-making effectiveness regarding unstructured or semistructured business problems

Capabilities of a Decision Support System

- Support for problem-solving phases:
  - A specific DSS might support only one or a few phases
- Support for various decision frequencies:
  - *Ad hoc DSS* is concerned with situations or decisions that come up only a few times
  - *Institutional DSS* handles situations or decisions that occur more than once
- Support for various problem structures:
  - *Highly structured problems* are straightforward, requiring known facts and relationships
  - *Semistructured* or *unstructured problems* are more complex
- Support for various decision-making levels:
  - DSSs can provide help for managers at various levels within the organization

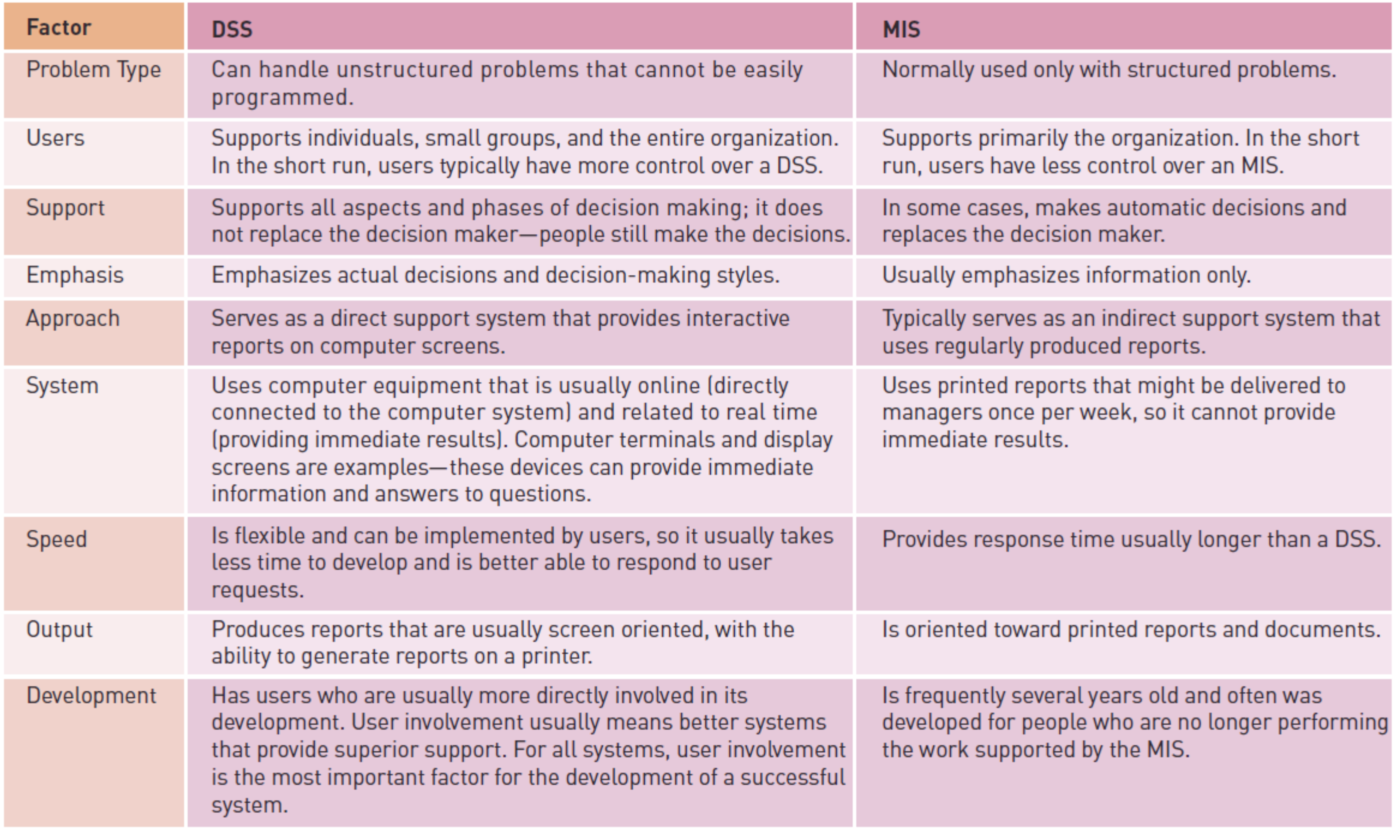
Decision-Making Level


Strategic
Tactical
Operational
Strategic managers involved with long-term decisions
Operational managers involved with daily decisions
High
Low
Decision Frequency

A Comparison of DSS and MIS

- DSS differs from an MIS in numerous ways, including:
  - The type of problems solved
  - The support given to users
  - The decision emphasis and approach
  - The type, speed, output, and development of the system used

A Comparison of DSS and MIS

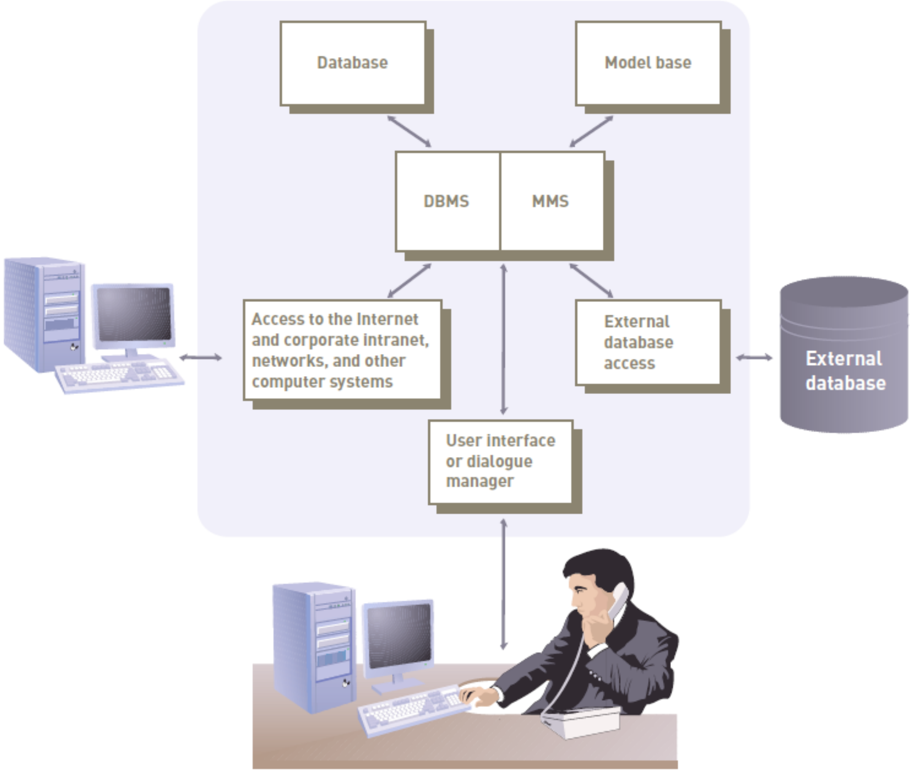


Factor	DSS	MIS
Problem Type	Can handle unstructured problems that cannot be easily programmed.	Normally used only with structured problems.
Users	Supports individuals, small groups, and the entire organization. In the short run, users typically have more control over a DSS.	Supports primarily the organization. In the short run, users have less control over an MIS.
Support	Supports all aspects and phases of decision making; it does not replace the decision maker—people still make the decisions.	In some cases, makes automatic decisions and replaces the decision maker.
Emphasis	Emphasizes actual decisions and decision-making styles.	Usually emphasizes information only.
Approach	Serves as a direct support system that provides interactive reports on computer screens.	Typically serves as an indirect support system that uses regularly produced reports.
System	Uses computer equipment that is usually online (directly connected to the computer system) and related to real time (providing immediate results). Computer terminals and display screens are examples—these devices can provide immediate information and answers to questions.	Uses printed reports that might be delivered to managers once per week, so it cannot provide immediate results.
Speed	Is flexible and can be implemented by users, so it usually takes less time to develop and is better able to respond to user requests.	Provides response time usually longer than a DSS.
Output	Produces reports that are usually screen oriented, with the ability to generate reports on a printer.	Is oriented toward printed reports and documents.
Development	Has users who are usually more directly involved in its development. User involvement usually means better systems that provide superior support. For all systems, user involvement is the most important factor for the development of a successful system.	Is frequently several years old and often was developed for people who are no longer performing the work supported by the MIS.

Components of a Decision Support System

- At the core of a DSS are a database and a model base
- Dialogue manager:
  - Allows decision makers to easily access and manipulate the DSS and to use common business terms and phrases

Conceptual Model of a DSS


Database
Model base
DBMS
MMS
Access to the Internet and corporate intranet, networks, and other computer systems
External database access
External database
User interface or dialogue manager

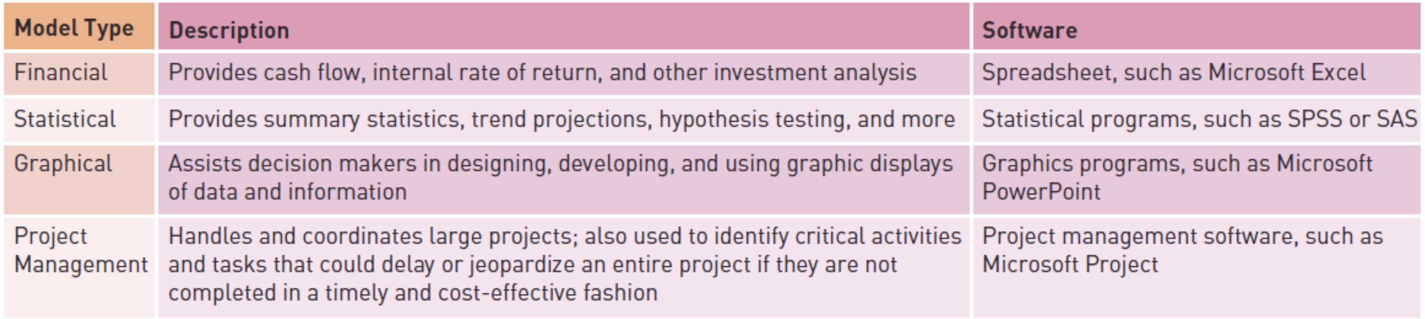
The Database

- Database management system:
  - Allows managers and decision makers to perform qualitative analysis on data stored in company’s databases, data warehouses, and data marts.
  - Can also be used to connect to external databases
- Data-driven DSS:
  - Performs qualitative analysis based on the company’s databases

The Model Base

- Model base:
  - Allows managers and decision makers to perform *quantitative analysis* on both internal and external data
- Model-driven DSS:
  - Performs mathematical or quantitative analysis
- Model management software (MMS):
  - Coordinates the use of models in a DSS

Model Management Software



Model Type	Description	Software
Financial	Provides cash flow, internal rate of return, and other investment analysis	Spreadsheet, such as Microsoft Excel
Statistical	Provides summary statistics, trend projections, hypothesis testing, and more	Statistical programs, such as SPSS or SAS
Graphical	Assists decision makers in designing, developing, and using graphic displays of data and information	Graphics programs, such as Microsoft PowerPoint
Project Management	Handles and coordinates large projects; also used to identify critical activities and tasks that could delay or jeopardize an entire project if they are not completed in a timely and cost-effective fashion	Project management software, such as Microsoft Project

An Overview of Artificial Intelligence

- Artificial intelligence (AI):
  - Computers with the ability to mimic or duplicate the functions of the human brain
- Many AI pioneers:
  - Predicted that computers would be as “smart” as people by the 1960s

Artificial Intelligence in Perspective

- Artificial intelligence systems:
  - Include the people, procedures, hardware, software, data, and knowledge needed to develop computer systems and machines that demonstrate characteristics of human intelligence

The Nature of Intelligence

Turing Test:

- Determines whether responses from a computer with intelligent behavior are indistinguishable from those from a human being

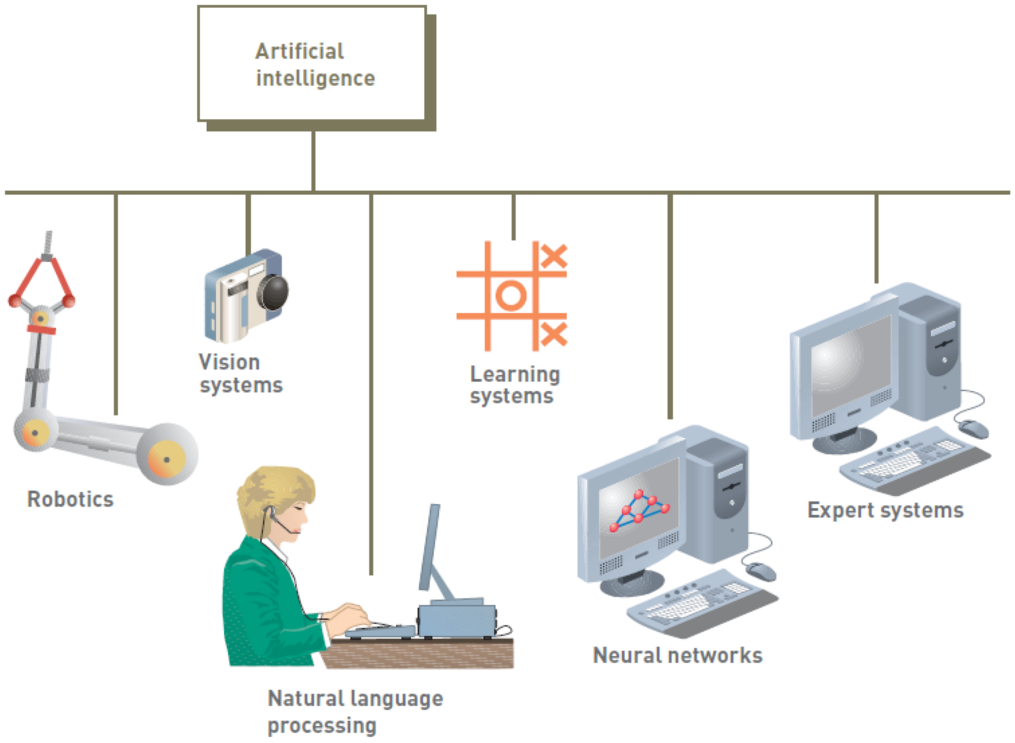
Characteristics of intelligent behavior include the ability to:

- Learn from experiences and apply knowledge acquired from experience
- Handle complex situations
- Solve problems when important information is missing
- Determine what is important
- React quickly and correctly to a new situation
- Understand visual images
- Process and manipulate symbols
- Be creative and imaginative
- Use heuristics

The Major Branches of Artificial Intelligence

- AI is a broad field that includes:
  - Expert systems and robotics
  - Vision systems and natural language processing
  - Learning systems and neural networks
- Expert systems:
  - Hardware and software that stores knowledge and makes inferences, similar to a human expert

A Conceptual Model of Artificial Intelligence



Robotics

- Developing mechanical devices that can:
  - Paint cars, make precision welds, and perform other tasks that require a high degree of precision
- Manufacturers use robots to assemble and paint products
- Contemporary robotics:
  - Combine both high-precision machine capabilities and sophisticated controlling software

Vision Systems

- Hardware and software that permit computers to capture, store, and manipulate visual images and pictures
- Effective at identifying people based on facial features

Natural Language Processing and Voice Recognition

- Processing that allows the computer to understand and react to statements and commands made in a “natural” language, such as English
- Voice recognition:
    - Converting sound waves into words

Learning Systems

- Combination of software and hardware that:
  - Allows the computer to change how it functions or reacts to situations based on feedback it receives
- Learning systems software:
  - Requires feedback on results of actions or decisions

Neural Networks

- Computer system that simulates functioning of a human brain
- Can process many pieces of data at the same time and learn to recognize patterns
- Neural network program:
  - Helps engineers slow or speed drilling operations to help increase drilling accuracy and reduce costs
- Some of the specific abilities of neural networks include the following:
  - Retrieving information even if some of the neural nodes fail
  - Quickly modifying stored data as a result of new information
  - Discovering relationships and trends in large databases
  - Solving complex problems for which all the information is not present

Other Artificial Intelligence Applications

- Genetic algorithm:
  - Approach to solving complex problems in which a number of related operations or models change and evolve until the best one emerges
- Intelligent agent:
  - Programs and a knowledge base used to perform a specific task for a person, a process, or another program

What is KNIME?

KNIME stands for *Konstanz Information Miner*. It’s an open-source platform for data analysis, integration, and visualization. The name comes from the University of Konstanz, where the project was originally developed, and it reflects the idea of exploring and mining information from data.

What makes KNIME special is that it allows users to create data processing workflows without writing code. Instead, it uses a visual, drag-and-drop interface where you can easily connect different steps together – for example, importing, cleaning, transforming, and analysing data.

In terms of applications, KNIME is widely used for:

Data cleaning and preprocessing,

Statistical analysis,

Building machine learning and AI models, and

Connecting with business intelligence systems such as Power BI.

It also has several key advantages:

First, it’s open-source, which means it’s completely free to use and can be extended by the community.

Second, it has a modular architecture with thousands of ready-made nodes that make complex workflows much easier to build.

And third, KNIME integrates very well with other tools like Power BI, Python, and R.

In short, we can say that KNIME is a flexible open-source platform that helps transform raw data into value – visually and efficiently.”

What is KNIME?

Power BI is a great tool for creating dashboards, reports, and visualizing business data in a clear and interactive way. However, Power BI is not primarily designed for heavy data preparation or complex machine learning tasks.

- KNIME can handle the entire data preparation and analysis process before the data is imported into Power BI.
- For example, KNIME can:
  - Connect to multiple data sources at once – such as Excel files, databases, or APIs.
  - Clean and transform large or messy datasets.
  - Combine, filter, and enrich data automatically.
  - Build and train machine learning or predictive models.
- Once the data is ready, KNIME can export the processed results directly to Power BI, where users can easily create dashboards and visual reports.
- In this way, KNIME acts as a powerful backend engine for Power BI. It extends Power BI's capabilities by adding advanced data analytics, automation, and predictive modelling.
- At the same time, Power BI provides the strong front-end visualization layer, turning the results from KNIME into clear and interactive insights.
- So, together, they form a complete analytics solution — KNIME for data processing and modelling, and Power BI for visualization and decision support.

Getting Started with KNIME

Where to get KNIME?

- Official website: https://www.knime.com
- The open source Knime Analytics Platform is free of charge, the Knime Business Hub is paid version. The functionalities of the two products are the same, the Business Hub in addition includes a number of useful IT features for team collaboraion, workflow deployment and management and date warehousing.

Installation

- Works on **Windows, macOS, Linux**
- Simple installation package (no special configuration needed)
- After installation, ready to use immediately

Registration

- **Not mandatory** for basic use

With a free account you get access to:

- **KNIME Hub (KNIME Hub - https://hub.knime.com/** → workflows, extensions, sharing
  - Access to shared workflows and nodes
  - Ability to publish your own workflows
  - Community support and collaboration
  - Cloud storage for workflows
- **KNIME Learning Center** https://knime.learnupon.com/my-learning → training materials, courses
- **K-AI assistant** → built-in AI support for building workflows

Power BI and Knime

KNIME and Power BI do not compete – they complement each other. Together, they cover the full data lifecycle: from raw data to actionable insights.

Power BI – Strengths

- Excellent for **data visualization and dashboards**
- Strong **business intelligence** features
- Easy integration with Microsoft ecosystem
- Fast insights for **decision support**

KNIME – Strengths

- Excellent for **data preprocessing and cleaning**
- Advanced **machine learning & AI integration**
- Handles **large, complex datasets** flexibly
- Open-source with thousands of **extensions and nodes**

Key idea:

- Power BI is **best at communicating insights** (visuals, dashboards, reports).
- KNIME is **best at preparing and enriching data** (preprocessing, modelling, automation).

Power BI and Knime

One of KNIME’s greatest advantages compared to Power BI is that it is not only a data transformation tool but also a true process orchestration platform. This means that users are not limited to transforming data but can also *control the entire dataflow*: **repeated executions, conditional branches, error handling, retries, alternative paths, and automatic logging** can all be implemented. While Power Query is primarily a transformation tool that works well if all the data is in place, KNIME ensures that workflows run fault-tolerantly and in a controlled way, so that errors do not break the entire process.

Power BI and Knime

A typical business example is the daily sales reporting pipeline. In practice, it often happens that the expected file (e.g. sales_20250908.csv) does not arrive on time. In Power Query this causes the refresh to fail completely. In KNIME, however, the Try/Catch nodes allow the workflow to first attempt to load the file, and if it is missing, the “Catch” branch automatically switches to another source, for example yesterday’s backup database or a placeholder record with zero sales. This way, the Power BI dashboard does not collapse but instead clearly indicates that data is missing, while still providing up-to-date reporting.

Power BI and Knime

Another common scenario is when monthly files need to be merged. Suppose there are 12 CSV files, one for each month, and one of them has a broken header. In Power Query this will often break the entire query, but in KNIME a Loop can be used to process each file individually. If an error occurs for a particular file, the Catch branch takes over and can retry with looser settings, repair the columns, or fill in missing values. The other 11 files are processed without interruption, while the problematic file is logged and flagged, and the user can automatically receive a notification.

Power BI and Knime

A third useful pattern is the introduction of quality gates. For example, if some records contain missing product IDs or future dates, the Rule Engine node can be used to tag data as “OK” or “BAD”. The “OK” records flow further into the reporting pipeline and into Power BI, while the “BAD” records are redirected to a quarantine table and automatically trigger an alert to the responsible data steward. This ensures that poor-quality data does not contaminate the reports, while still being tracked and available for later correction. These solutions are made possible by KNIME’s visual control elements such as Try/Catch, various Loop nodes, the Row Splitter and Empty Table Switch, as well as logging and notification nodes such as Send Email. With these, KNIME workflows are more reliable, flexible, and easier to maintain than refresh-based solutions in Power BI.

Power BI is excellent for visualization and self-service analytics, while KNIME ensures robust, enterprise-level data preparation and orchestration. Together, they form a system where KNIME performs the complex data and AI work, and Power BI serves as the master of visualization and decision support.

What is the difference between Power BI and KNIME in machine learning?

- Power BI AutoML: it is quick and easy, but essentially a black box. You choose whether you want prediction or classification, the system runs a few algorithms in the background, and you get a result. It’s a good entry point, but you have very little control: you cannot decide which algorithms are tested, you cannot fine-tune the parameters, and the whole training process is hidden.
- KNIME ML/MLOps: it provides a fully visual, end-to-end data science pipeline. You don’t just build a model; you can control how it learns, which parameters it tries, how it evaluates itself, and how it gets deployed into production.

Machine Learning in KNIME vs. Power BI

1. Cross-validation (CV)

What it is: You split your dataset into several parts (e.g. 10 folds). The model is trained on 9 parts and tested on the remaining 1, and this process repeats until all parts have been tested.

Why it matters: It prevents overfitting (when a model learns the training data too well but fails on new data).

Example: You have 1,000 customer records. With 10-fold CV, the model is trained on 900 customers and tested on 100, repeated 10 times with different splits.

In KNIME: The X-Partitioner and X-Aggregator nodes make the entire CV process transparent and reproducible.

In Power BI AutoML: Cross-validation happens behind the scenes with no user control.

Machine Learning in KNIME vs. Power BI

2. Hyperparameter tuning

What it is: Every algorithm has parameters (“knobs”) that influence performance. Hyperparameter tuning means finding the best values for these knobs.

Why it matters: Wrong settings lead to weak models; tuning can dramatically improve accuracy.

Example: A decision tree can be shallow (fast but inaccurate) or very deep (accurate but prone to overfitting). Tuning finds the optimal depth.

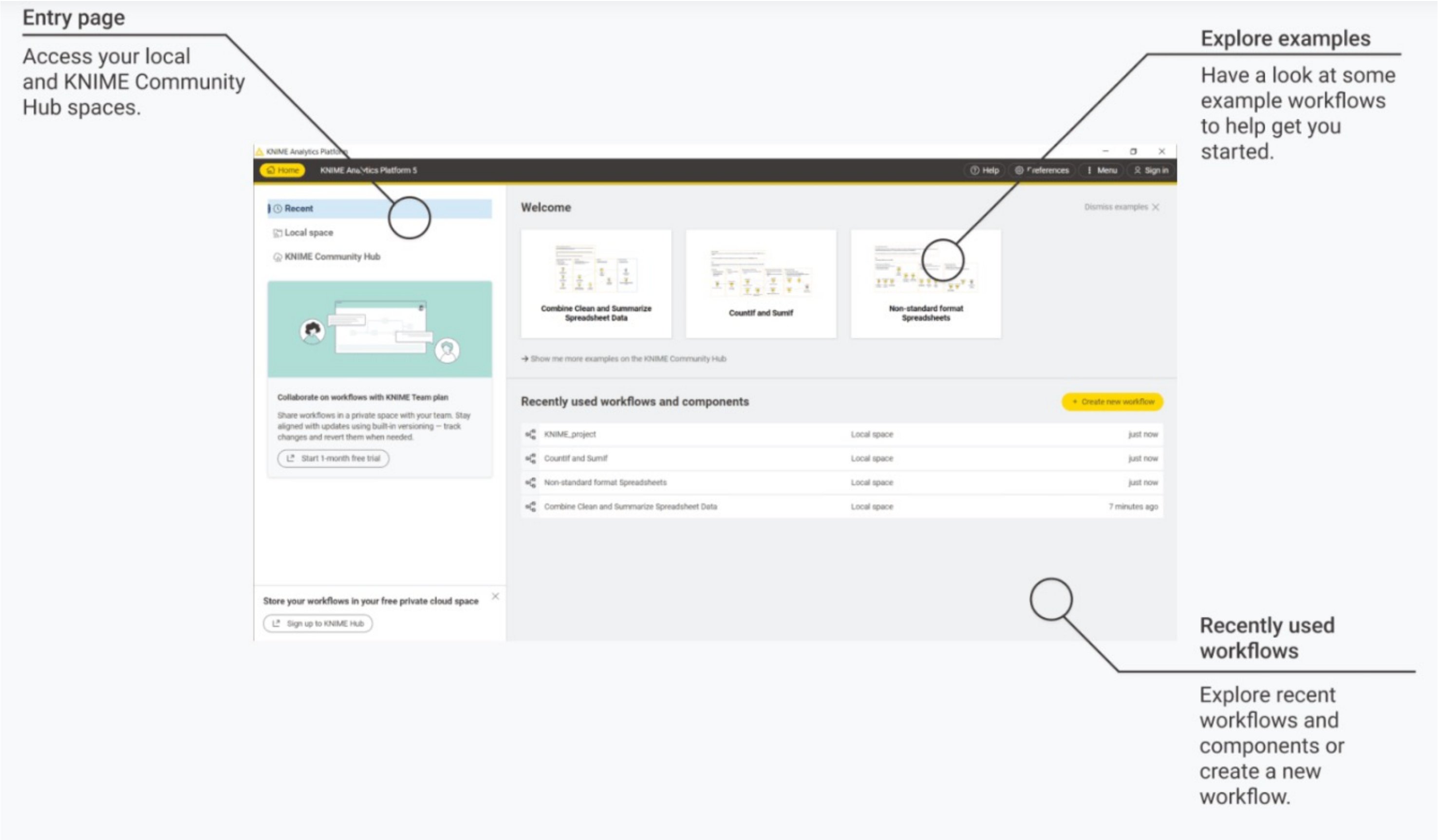
In KNIME: You can use the Parameter Optimization Loop Start/End nodes, which systematically try different parameter combinations and choose the best one.

In Power BI AutoML: Parameters are fixed or hidden; users cannot fine-tune.

Machine Learning in KNIME vs. Power BI

3. AutoML components
What it is: Ready-made workflows that automatically test multiple algorithms and rank them.
Why it matters: Saves time, offers transparency, and helps non-experts build strong models.
Example: With churn prediction data, an AutoML component in KNIME will automatically test Logistic Regression, Random Forest, Gradient Boosted Trees, and Neural Networks, then output the best-performing model.
In KNIME: AutoML components are downloadable from the KNIME Hub and customisable.
In Power BI AutoML: AutoML exists but as a “black box”—users cannot see how models are compared.

KNIME Interface – Start Pages



Entry page

Access your local and KNIME Community Hub spaces.

Explore examples

Have a look at some example workflows to help get you started.

Recently used workflows

Explore recent workflows and components or create a new workflow.

Collaborate on workflows with KNIME Team plan

Share workflows in a private space with your team. Stay aligned with updates using built-in versioning – track changes and revert them when needed.

Store your workflows in your free private cloud space

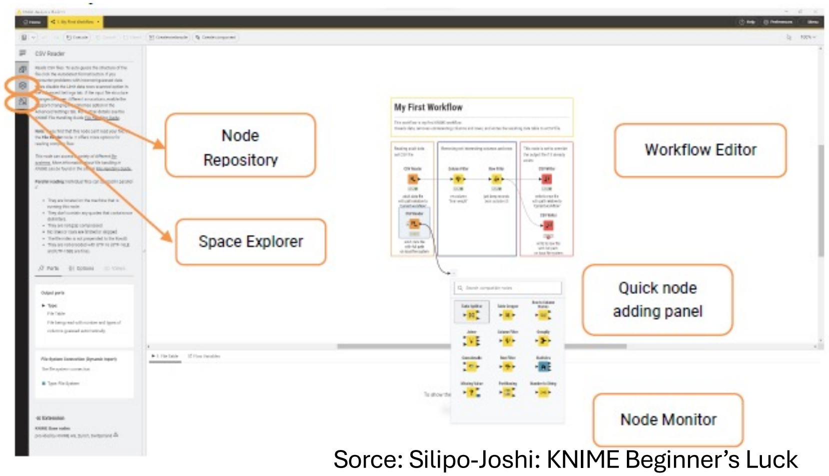
Recently used workflows and components

KNIME_project	Local space	just now
CountIf and SumIf	Local space	just now
Non-standard format Spreadsheets	Local space	just now
Combine Clean and Summarize Spreadsheet Data	Local space	7 minutes ago

Sorce: Silipo-Joshi: KNIME Beginner’s Luck

KNIME Interface – Start Page

- Left panel: Home, Recent, Local Space
- Create workflow groups (folders)
- Export / Import workflows
- Connect to KNIME Community Hub
- Help, documentation, keyboard shortcuts



Sorce: Silipo-Joshi: KNIME Beginner’s Luck

KNIME Interface – Start Pages

Application tabs

Entry page tab and all opened workflows tabs.

Side panel navigation

Description
Description of the displayed workflow or component.

—

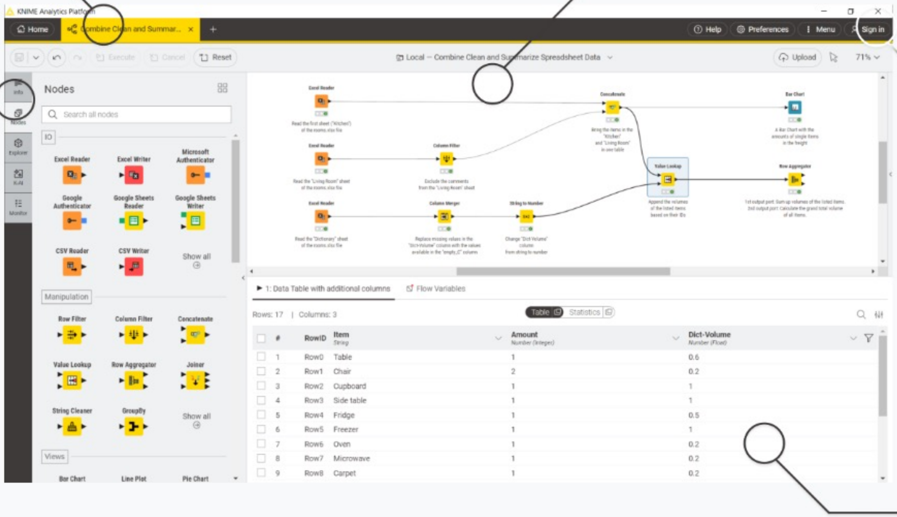
Node repository
All available nodes in KNIME Analytics Platform to build your workflows.

—

Space explorer
Navigate local or KNIME Hub spaces and access your workflows, components and files.

—

Workflow monitor
Debug your workflows by checking which node, component, or metanode has errors or warnings.


KNIME Analytics Platform
Home
Combine Clean and Summar...
+
Help
Preferences
Menu
Sign in
Execute
Cancel
Reset
Local – Combine Clean and Summarize Spreadsheet Data
Upload
71%
Info
Nodes
Explorer
K-AI
Monitor
Nodes
Search all nodes
IO
Excel Reader
Excel Writer
Microsoft Authenticator
Google Authenticator
Google Sheets Reader
Google Sheets Writer
CSV Reader
CSV Writer
Show all
Manipulation
Row Filter
Column Filter
Concatenate
Value Lookup
Row Aggregator
Joiner
String Cleaner
GroupBy
Show all
Views
Bar Chart
Line Plot
Pie Chart
Excel Reader
Read the first sheet ("Kitchen") of the rooms.xlsx file
Excel Reader
Read the "Living Room" sheet of the rooms.xlsx file
Column Filter
Exclude the comments from the "Living Room" sheet
Excel Reader
Read the "Dictionary" sheet of the rooms.xlsx file
Column Merger
Replace missing values in the "Dict-Volume" column with the values available in the "empty_C" column
String to Number
Change "Dict-Volume" column from string to number
Concatenate
Bring the items in the "Kitchen" and "Living Room" in one table
Value Lookup
Append the volumes of the listed items based on their IDs
Bar Chart
A Bar Chart with the amounts of single items in the freight
Row Aggregator
1st output port: Sum up volumes of the listed items. 2nd output port: Calculate the grand total volume of all items.
1: Data Table with additional columns
Flow Variables
Rows: 17 | Columns: 3
Table
Statistics

□	#	RowID	Item String	Amount Number (Integer)	Dict-Volume Number (Float)
□	1	Row0	Table	1	0.6
□	2	Row1	Chair	2	0.2
□	3	Row2	Cupboard	1	1
□	4	Row3	Side table	1	1
□	5	Row4	Fridge	1	0.5
□	6	Row5	Freezer	1	1
□	7	Row6	Oven	1	0.2
□	8	Row7	Microwave	1	0.2
□	9	Row8	Carpet	1	0.2

Workflow editor

Canvas for editing the currently active workflow.

Help, Preferences, Menu

Access to more material, install additional extensions and change settings for the node repository.

Node monitor

Shows the output of the current selected node and also the flow variable values.

Sorce: Silipo-Joshi: KNIME Beginner’s Luck

Left side panel

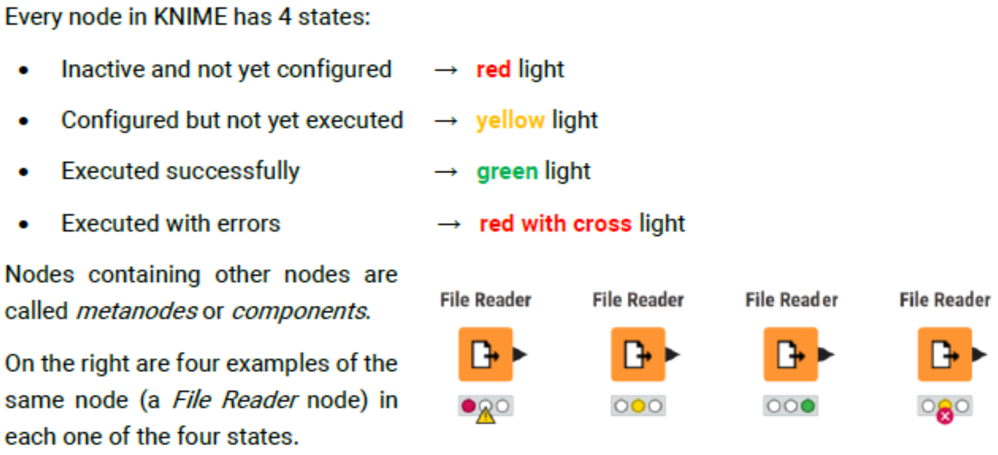
- Explorer panel – file system view
- Monitor panel – errors and warnings
- AI Assistant – workflow generation, Q&A
- Add annotations to canvas
- Zoom, Pan, Select modes in editor

Basic elements

- What is a **Workflow**? A data analysis sequence, which in a traditional programming language would be implemented by a series of instructions and calls to functions.
- What is a **Node**? A node is the single processing unit of a workflow
- **Metanode** is simply a container that helps organize and tidy up workflows—it groups nodes together, making complex workflows easier to read and maintain, but metanodes cannot have custom configuration dialogs or views, and all variables pass in and out without restrictions..
- **Component** is a reusable, configurable part of a workflow that can have its own configuration dialog, custom interactive views, and local variable scope (variables inside a component do not affect the outside by default). Components are great for sharing or reusing functionality and for hiding workflow complexity.

Workflow Editor (Canvas)

- Central area: workflow editor (canvas)
- Left panel: Nodes panel (I/O, Data Manipulation, Views, Machine Learning, Database, etc.)



Every node in KNIME has 4 states:

- Inactive and not yet configured → red light
- Configured but not yet executed → yellow light
- Executed successfully → green light
- Executed with errors → red with cross light

Nodes containing other nodes are called *metanodes* or *components*.

On the right are four examples of the same node (a *File Reader* node) in each one of the four states.

File Reader
File Reader
File Reader
File Reader

Sorce: Silipo-Joshi: KNIME Beginner’s Luck

Searching and Describing Nodes – Node Repository

- Fuzzy search bar
- Filter by category (e.g., I/O, ML)
- Open node description (question mark icon)
- Check ports, outputs, parameters

Configuring a Node – Table Creator Example

- Create sample table manually or paste data
- Set column names and data types (integer, string, etc.)
- Must re-run after configuration changes
- View output after execution

Filter columns

https://hub.knime.com/knime/spaces/Education/Educational%20Videos%20Demo%20Workflows/How%20to%20Filter%20Columns~5hV0KwdBhDYWEZv_/current-state

How to filter columns from a table in KNIME Analytics Platform in three different ways: manual, pattern based or type based selection.

You have a table of sales data. It contains the country and the date of the sale, a column card indicating whether the transaction has been performed with credit card or not, and finally various columns with the IDs of the products involved in the transaction.

Filter Rows

- https://hub.knime.com/knime/spaces/Education/Educational%20Videos%20Demo%20Workflows/How%20to%20Filter%20Rows~vmRuAu1mWi95QEGH/current-state
- You have a sales table, where each row contains the data for a sales contract. There is the sold product, the country of sale, the date of the contract, the number of items, the amount of money generated by this contract, and whether the transaction was paid by credit card or not.
- Some columns are of type String, some columns are of type Integer. Some columns have missing values, like here, for example.

How to handle missing values

https://hub.knime.com/knime/spaces/Education/Educational%20Videos%20Demo%20Workflows/How%20to%20Handle%20Missing%20Values~8U0NA6i5hTH0EnJo/current-state

When dealing with real data, it is common that some values are missing for a certain column. In KNIME Analytics platform, missing values in a table are represented by the red question mark. For example, in this table of customers, there are missing values in the columns CustomerID, Age, Email and so on. You can handle missing values with the Missing Value node.

In its configuration, you can select a default replacement for all the missing values of a certain type. For example, you can place a fixed value such as the string 'unknown' every time a string is missing.

In some cases, you want to be more granular. For example, a customer with a missing customerID is not useful, hence you want to remove the corresponding row. To do that, use the Column Settings tab. Double click the column name CustomerID and select Remove Row. All the rows with missing customerIDs are now excluded from the output table. Note that the replacement options vary depending on the column type.

How to handle missing values

To handle the missing values of the column Age, which is an integer column. In this case, you can choose to replace a missing value with a statistics computed over the values in the column, such as the maximum, the mean, or the most frequent value and soon. Once you define a strategy to handle missing values, you can apply it to new data using the Missing Value (Apply) node.

This node needs no configuration.

It just applies the same missing value strategy to another data table.

In this case, it removes the rows with missing customerIDs and uses the mean age when the age is missing.

Note that for statistical values— such as the mean, most frequent value, etc.— the Missing Value (apply) will use the statistic of the original table and will not recalculate it for the new table.

How to handle missing values

In some cases, information is spread or missing over multiple columns. For example, customers have an Email or a corporate email, or both. For you, one email is enough, no matter if personal or corporate. You can merge those two columns with a Column Merger node.

Select Email as the primary column, and populate its missing values with the values of the secondary column Corporate Email.

Finally, some columns have so many missing values that they are of no use.

You can choose to remove them by using the Missing Value Column Filter node.

Select the columns to test, for example, Newsletter, and set a threshold.

Since the Newsletter column has more than 60% of rows with missing values, it is excludedfrom the table.

How to split cells

https://hub.knime.com/knime/spaces/Education/Educational%20Videos%20Demo%20Workflows/How%20to%20Split%20Cells~aG7C3M94PVbXwxdE/current-state

Sometimes, data loves to stick together. No matter if the result of a computation or data is not saved in a clear way, sometimes a cell contains more than one information and it is more handy to keep it separate.

This table contains three columns, each storing multiple information.

For example, the column CustomerID stores a string with two pieces of information, the Group and the actual customerID, separated by an underscore. To divide it, add a Cell Splitter node. Select the column CustomerID and enter the delimiter “underscore”.

With the default configuration, the node splits the content of the column into multiple columns for each occurrence of the divider. In this case, you get one column containing the group id and one with the customer id.

How to split cells

Consider now the second column. The country and city are separated by a space. Add a new cell splitter node and type a space in the delimiter field.

Note that the table in output is not correct, since some cities also contain spaces within their names. In this case, you are lucky, since city names are enclosed in quotation marks. Set the quotation mark character in the configuration window of the Cell Splitter. This will make it skip all the space delimiters enclosed in quotation marks.

For more complex cases, for example if those words were not enclosed in quotation marks, you can explore more sophisticated solutions with regular expressions, that can be applied with other nodes such as the Regex Split.

If the column name can be split in the same manner as the column's content, you can select the checkbox "split input column name". In this case, the node renames the splitted columns with the splitted column names.

How to split cells

Finally, the Cell Splitter node also allows you to create collection columns. Collections is the column type to use if you want to correctly store multiple elements in the same cell.

Select the column “Product”, set the delimiter “comma” and set the output

“As list”. The output table contains a new column of type “List” with the content of the column stored in an array format. You can also select “As Set”, if you want to remove all the duplicate elements from each cell.

Concacenate tables

https://hub.knime.com/knime/spaces/Education/Educational%20Videos%20Demo%20Workflows/How%20to%20Concatenate%20Tables~uKGwDVUOztlLErJY/current-state

With the concatenate operation you can merge two or more tables by putting them on top of each other.

The common columns appearing in both input tables will always show in the output table.

If exclusive columns are appearing in only one of the tables, you have two ways to proceed:

take the intersection of the columns and retain only the common columns, or the union of columns, and retain all the columns in both tables regardless of whether they are common or exclusive. Have a look at these two tables. They have one exclusive column each - “2” and “3”. If you concatenate the tables and choose to keep an intersection of columns, the output table will not contain these exclusive columns.

If you choose to keep a union of columns instead, these columns will appear in the resulting table together with the common columns.

Note that the concatenation operation retains the row order.

Concacenate tables

The tables of this workflow hold sales information. The first dataset contains sales records for the years 2010 and 2011, with an exclusive column “card” unique to this dataset. The second dataset contains sales records for 2011, with an exclusive column “insurance” unique to this dataset. To put together these two tables, you can use the Concatenate node. First, select to combine the input columns by intersection.

The node also lets you select how to handle duplicate RowIDs. Choose to append a suffix. Close the configuration and execute the node. As expected, the exclusive columns “card” and “insurance” are not there. In addition to that, you can see some row IDs with a duplicate suffix, indicating duplicate row IDs in the input tables.

Select now to keep the union of columns. The “card” and “insurance” columns are now included. The “insurance” column is filled with missing values for the rows originating from the first data set. Likewise missing values are seen in the “card” column for the rows from the second table. This is because these columns appear only in one of the datasets. Finally, notice that the Concatenate node has dynamic ports to let you concatenate as many tables as you want.

Simply drag the connection from another table to create a new input port.

You can also add input and output ports to nodes by clicking the plus button.

What is a loop?

How to build a loop, and how to distinguish between different types of loop nodes in KNIME Analytics Platform.

Imagine the following use case: you have a list of orders from different countries and you want to write the orders for each country into a separate sheet, in this example an Excel sheet. What you can do, of course, is to check which countries are there in the table. Select the first one. Filter its orders. Write it to a separate sheet,   and then repeat the process for the next country, until you go through all of them. This can be a tedious and error prone process. What if the number of countries is large? Or, what if new countries appear in the orders from time to time? Is there a simpler way?

https://hub.knime.com/knime/spaces/Education/Educational%20Videos%20Demo%20Workflows/What%20is%20a%20Loop~4HFN6xF3bPkO84r7/current-state

What is a loop?

A loop can iterate over a workflow segment and repeat its execution automatically for different inputs. What changes for each iteration can be a parameter value, a dataset, a subgroup of the same dataset, a single column, or a single row as flow variables.

A loop in KNIME begins with a Loop Start node, contains the loop body, and ends with a Loop End node.

Generally, the Loop Start node increases the iteration counter and sends the data to the loop body. The loop body then executes the operations to be repeated. After those are performed the Loop End node checks if the end condition is fulfilled, and if not, the Loop Start node starts the next iteration and the loop body performs the operations again. When the end condition is fulfilled, the Loop End node collects the data from the different iterations and closes the loop.

What is a loop?

Let's look at a simple example of a loop that can solve the task discussed earlier, namely to filter the orders for each country and write them to a separate sheet.

The Group Loop Start node here defines the column categories the loop iterates through, **At each new iteration, it filters the data for one category,** and sends it to the loop body. It also exports a flow variable describing which category is used in this iteration. In the Loop Body, we have only one node – the Excel Writer- which, at each iteration, writes the order data from the country to a separate sheet. This node will name each sheet according to the corresponding country, because it is parametrized by the category flow variable from the Group Loop start node. The loop body can, of course, be as complex as you need - with multiple nodes, more advanced workflow structures, and even nested loops. Finally, the Variable Loop End node checks whether the loop iterated through all the countries, and if yes, closes the loop and collects flow variables from all the iterations.

What is a loop?

For different types of loops, there are different Loop Start and Loop End nodes. For example, A Group Loop Start you saw in the example, iterates over groups of data until it iterates through all of them. A similar node is a Chunk Loop Start - it does the same for consecutive data chunks of a specified size. These two nodes are useful when in each loop iteration, you need to process a different group of data.

Another common node is a Counting Loop Start node that performs a specified number of iterations. This node is useful when you know beforehand how many times the loop should execute. A different type of loop is a recursive loop.

A Recursive Loop Start node iterates over the loop body output data table until a particular condition is met. This loop is useful, for example, when you need to retry some action until success.

You can see that there are many other loop start and end nodes available in KNIME Analytics Platform. While some loop start nodes can be paired only with one type of loop end node, some loop start nodes have several options with which loop end node to be combined.

Build, Excecute and Debug loops

https://hub.knime.com/knime/spaces/Education/Educational%20Videos%20Demo%20Workflows/Build,%20Execute,%20and%20Debug%20a%20Loop~iHKbPUi4RvfPT_0B/current-state

This workflow demonstrates how to use a loop to create a timestamp on the 15th for each month of the year 2023. We start from day = 15 and year = 2023 in Table Creator and we insert the current iteration number 12 times to simulate the month. The resulting data table has 12 data rows in the shape 15/MM/2023.

Some Loop Start nodes have dynamic ports, meaning you can add different types and numbers of ports. If you use the quick node adding panel, the correct port will be added automatically.

We set the number of iterations to 12.

Each Loop Start node produces the flow variable "currentiteration" with the counter of the current iteration.

Counting in KNIME Analytics Platform starts from 0, so the first iteration number is 0.

In the loop body we then use a Math Formula node to add 1 to the current iteration number and produce the month value.

Build, Excecute and Debug loops

String Manipulation node to create a timestamp.

Finally, we add a Loop End node. In the configuration of this node, you can: decide how to treat duplicate row IDs from different iterations

The most straightforward way to execute the whole loop is by executing the Loop End node.

While the loop is executing, you can see the progress loop sign at the Loop End node.

At the node monitor, you can see the progress of the loop execution in the changing flow variable “currentiteration” and the maximum number of iterations.

Build, Excecute and Debug loops

The resulting table has 12 rows with the respective timestamp and the iteration number in which each row was created.

If you want to reset the whole loop, you can reset any node in the loop body as well as the loop start and the loop end node.

Besides, sometimes a loop can fail and you might need to debug a particular iteration. This is where the step-wise execution comes in handy. For that, in the Loop End node action bar, click the “Step” button.

This will execute one iteration of the loop and pause. Another way to debug a loop is by using a Breakpoint node.

It can be inserted in any place in the loop body and, when enabled, halts execution when the incoming data fulfills a specified condition. A breakpoint can activate, for example, when a variable matches a particular value. For instance, we want to stop the loop when it reaches June, meaning at iteration number 5.

Flow variables

In this workflow, we read in some sales data, filter the entries for Germany, calculate the total revenue, and finally, rename the total revenue column to “Germany”.

This workflow works fine for the country of Germany. However, if you want to obtain the same table for the USA, you need to change the configuration of both the Row Filter and the Column Renamer nodes.

This is a simple workflow. But if you had a more complex workflow with more nodes and settings to update - it would be inefficient and error prone to reconfigure it manually. This is exactly where you need flow variables.

https://hub.knime.com/knime/spaces/Education/Educational%20Videos%20Demo%20Workflows/Creating%20and%20Using%20Flow%20Variables~FTdItnkkVbDeb1sf/current-state

Flow variables

A flow variable is a parameter in a KNIME workflow to overwrite a node's configuration settings.

In our example you can control node's settings configuring the country with a parameter - flow variable - to execute the workflow for different countries, without manually changing the settings.

Similar to data columns, flow variables can be of different types: string, integer, double, arrays, or Path. There are different ways to create a flow variable.

Have a look at how we can export a node's configuration as a flow variable. Now that we have defined the filtering pattern "Germany" in the Row Filter node, we can export it as a flow variable and reuse this pattern in the downstream nodes. Filtering pattern is a common setting to be controlled by a flow variable in the Row Filter node. Therefore, for this setting there is a special flow variable button to both control and export this setting. We can export it by clicking the checkbox and writing the name of the flow variable - "country".

Less common configuration settings don't have a special flow variable button. But all the settings can be controlled and exported in the Flow Variables tab. You can see that the pattern setting is here too.

Flow variables

In the node monitor, in the “Flow Variables” tab, you can now find the variable “country” with value “Germany”. Once created, flow variables are passed to all the downstream nodes in a workflow branch. Therefore, the “country” variable can be used by the subsequent nodes in the workflow.

Another way to create a flow variable is by using a dedicated node. This can be, for instance, configuration nodes, Table Row or Column to Variable nodes or the Variable Creator node.

The Variable Creator node is very simple:

It allows you to create multiple variables of different types. Table Row or Column to Variable nodes create a flow variable from the first row or column of the input table, respectively. Configuration nodes can be useful when you want to prompt a user to provide a variable of a particular type in a dedicated node or a component.

In our example, we need to create a flow variable of type String, therefore, we use the String Configuration node. We set the name of the flow variable to “country” and the default value to “Germany”. The red circle at the node output indicates a flow variable port and contains the created variable.

Value Lookup and Joiner

https://hub.knime.com/knime/spaces/Education/Educational%20Videos%20Demo%20Workflows/Value%20Lookup%20and%20Joiner~ujtZJXWlSD1YB-CR/current-state

Value lookup is an operation that lets you append values from a dictionary table to a data table, according to a matching column. In the data table you have the Order number and the Store ID.

In a second table, the "dictionary table", you have for each storeID, the information whether the store is an Online or Onsite store.

You can use the Value Lookup node to bring, to the data table, the information about the StoreType from in the dictionary table, according to the matching storeID.

Value Lookup and Joiner

In this example, the data table contains the orders, and the dictionary table contains the information about the stores. In the configuration window, select the two columns that have to match. In this case, the column containing the storeID in the data and dictionary tables. As you can see, the columns don't have to have the same name. You can also select which columns you want to retrieve from the dictionary table. In this case, select to only keep the StoreType column, since the StoreID column is already used as lookup.

In case of multiple matches, that is, if the dictionary column contains more rows with the same storeID, you can select to use the first or the last of these rows as they appear in the dictionary table.

For the last row, the node outputs a missing value for the store type, since the store_0000 is not available in the dictionary table. In case of no matches like this one, you can also select to match the next smaller or larger value of the dictionary.

Note that the node also offers advanced settings if you only want to match a substring of the lookup value, to match with wildcards and regexes, and to append an additional column indicating whether a match was found.

Note that the value lookup node covers frequent yet basic use cases. For more complex operations, such as using more than one matching column, have a look at the joiner node.

Row aggregators, Group by and Pivot

https://hub.knime.com/knime/spaces/Education/Educational%20Videos%20Demo%20Workflows/Row%20Aggregator,%20GroupBy%20and%20Pivot~mdX5LTIHLx5eaqvr/current-state

Let's see how to perform data aggregation in KNIME Analytics Platform using the Row Aggregator node.

This data table contains transaction data. Each order is repeated in multiple rows, each of which contains a single product of the order, its unit price and the quantity in that order. You want to calculate how many items are in each order. This means that you need to select all the rows of a certain OrderNumber and sum up the Quantity.

Drag & drop the Row Aggregator node and open its configuration window.

As Category Column, select the Order Number. You can choose among multiple Aggregation methods. Select Sum. Finally, select only Quantity as the Aggregation column. The output table contains the unique orders and the total amount of items in each of them.

You can also select a weight column when performing the aggregation. In this example, the total value of an order is given by the price of the products, multiplied by the quantity. Configure the Row Aggregator to perform this additional operation. Select the column "Price" as "Weight column". This time, while calculating the sum of item quantities for each order, the node multiplies each quantity for the respective price. Finally, by selecting "grand totals", the second output port of the node shows the total sum of the table values without the category column.

Row aggregators, Group by and Pivot

To aggregate the table in order to obtain the number of times that a certain customer bought a product. Therefore, you need to select each combination of product and customer. In the configuration window of the GroupBy node, in the Group tab, include the columns CustomerID and ProductNr. Executing the node as is will output all the combinations of customer and products available in the table.

To count the number of times an item was bought by a customer, you need to sum up the quantity column. To do that, move to the Manual Aggregation tab. Double click the column Quantity, and change the Aggregation Method to “Sum”. The node creates a new column containing the sum of the quantity for each combination of customer and product. The aggregation power of the GroupBy node does not stop here. You can add as many aggregation methods as you need: On the same column [Quantity + max] Or on multiple columns [Date + First] Note that the list of aggregation methods changes according to the data type of the aggregation column.

How many times a customer bought a product? Count for example the date. Among them, the "Count" aggregation method is the one that you can use to count the total number of records per group. This aggregation method is independent from the column values. That means "Count" returns the same value, no matter on which column you're applying the aggregation method. Note that for each aggregation method you can also set to include or ignore possible missing values, by ticking the corresponding checkbox.

Row aggregators, Group by and Pivot

- Pivoting

To know the number of times each product has been purchased in the Online and Onsite stores.

In the pivot table you want to report the product numbers in the rows and the StoreTypes in the columns. Each cell contains the total number of times a particular product has been purchased in that store type. Using the Pivot node in KNIME Analytics Platform, it is easy to create this kind of aggregated tables. The different values of one column become unique rows and the different values of another column become column headers. In addition, an aggregation method is applied to a third column. Every time you create a pivot table, you have to define which column to use to generate the rows, which column to use to generate the column headers, and which column to use to apply the aggregation method. The column "ProductNr" is the group column. This means, the values in this column generate the rows in the pivot table. The "StoreType" column is the pivot column. This means, the values in this column generate the column headers in the pivot table. "Quantity" is the column chosen for the aggregation and the aggregation method is sum.

In addition, you can define some more advanced settings in the lower part of the configuration dialog. Using these two menus you can define the column names of the pivot table.

In the first menu, select if you want to include the pivot column value and the aggregation method name or only the pivot column value. In the second menu, select how the aggregation method name is going to appear - whether it should be included in the column names or not.

In addition to the pivot table, the node has two other output tables. The second output table shows the group totals, that is, the totals by row. The third output table shows the pivot totals,

Convert String to Date&Time Data Type and Extract Date&Time Fields

https://hub.knime.com/knime/spaces/Education/Educational%20Videos%20Demo%20Workflows/Convert%20String%20to%20Date&Time%20Data%20Type%20and%20Extract%20Date&Time%20Fields~sGp08GeBZgOno7j4/current-state

This workflow shows examples of the following operations on date and time values:

converting date and time values from String to Date&Time; extracting granularities (years, months, etc.) from Date&Time values.

a data table has columns that contain date and time values. Yet they are often imported as Strings.

However, it is problematic for machines to read such strings with the same logic humans would do. The reason for this is that date&time values have different granularity - which includes date, time, and the time zone - and can be saved in varying order, or separated by different symbols.

Therefore, to be able to process date and time values, you need to convert strings to the dedicated Date&Time type, which enables machines to read these values correctly.

Convert String to Date&Time Data Type and Extract Date&Time Fields

In this workflow, we want to aggregate orders data by month for each year and visualize the results. For aggregation, we need year and month values of the orders. However, the “OrderDate '' column is of type string and we need to convert it to Date&Time data type.

In the configuration dialog of this (String to D 9node you first select the column to convert. In the next section, you can select the “Replace selected columns” option to replace the string column. Alternatively, you could append a converted column to the table.

Next, we want to calculate the yearly and monthly sales. For this we need to extract the date and time fields - in our case, years and months – from the “OrderDate'' column with the Extract Date&Time Fields node. In the configuration dialog, we first select the “OrderDate” column. Then you can select which date fields you want to extract from the order dates. In our case, those are: “Year”, “Month (number)”, and “Month (name)”. Finally, we set the language of the month names to “en-US” in this “Locale”

See the metanode configuration and the chart

Create and configure a componenet

https://hub.knime.com/knime/spaces/Education/Educational%20Videos%20Demo%20Workflows/Create%20and%20Modify%20a%20Component,%20Create%20Component%20Configuration%20and%20Composite%20View~bACuHb3OltXXOb5-/current-state

ssee a workflow that contains a configuration node. This Date&Time Configuration node allows you to create a flow variable with a date value. The Date&Time-based Row Filter then filters the data using this flow variable.

Start by selecting the nodes. Next, right click and select "Create Component". The selected nodes get wrapped into a component and you can directly give it a name, for example, "Filter Sales Data by a date".

Right click on the component and select "Component", you can see possible actions, such as open, rename, change the layout, expand, or share a component - locally or on KNIME Community or Business Hub. You can quickly open the component by pressing "Ctrl" and double clicking on it.

Create and configure a componenet

Configure the component in the same way as any other KNIME node. In the configuration dialog, the date input is shown as defined by the Date&Time Configuration node inside the component. If you change the value, the component will produce different results.

However, now you don’t see the output unless you open the component. Let’s modify it. First, let’s add an output table port to it. For that, hover over the component, click the plus button on the right side of it, and select Table port type. Next, let’s open the component and connect the table output to the component output.

Add other configuration settings to the component’s configuration dialog. Let’s add a Nominal Row Filter Configuration node to filter the data by country. In its configuration dialog, you can enter the component configuration setting label, name the output variable, select the column to filter data by and lock it, and select the default values. Now, if you check the component’s configuration dialog, the newly created configuration setting appears there as well. You can change the layout of the component configuration dialog. For that, open the layout editor, navigate to “Configuration Dialog Layout”, and drag and drop the configuration settings.

Create and configure a componenet

Another thing you can modify is the flow variables scope. By default, a variable defined inside a component is not available outside of it, and a variable defined outside the component is not available inside. To change the default behavior, inside of the component, in the Component Input, you can import variables from the main workflow to the component.

Similarly, in the Component Output, you can export variables from the component to the main workflow.

Try and catch, error handeling

https://hub.knime.com/knime/spaces/Education/Educational%20Videos%20Demo%20Workflows/Error%20Handling%20with%20Try-Catch~MYGfeUM2m0B6wHX4/current-state

To handle errors in KNIME workflows with Try and Catch block.

Here a table with new contracts is read from a remote data source, Google Sheets in this case, and is appended to a table with all the contracts. If we want new contracts to be added automatically on a regular basis, this workflow should run smoothly or get fixed as soon as possible in case of errors.

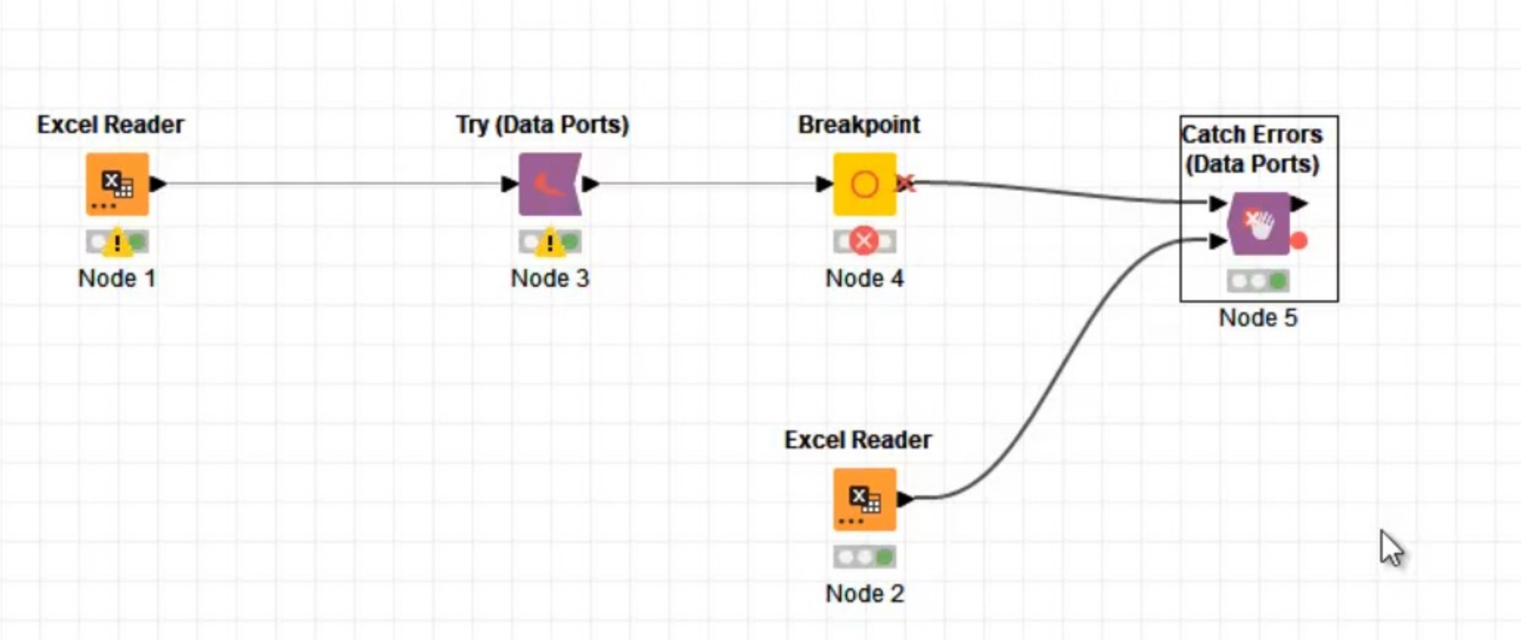
But imagine the Google Sheets Reader node unexpectedly fails due to some external reason, for example, due to the file not being there. If the workflow fails, the data isn't added, and no one is notified. This is where the Try and Catch block comes in handy.

Try and catch, error handeling

The Try and Catch block always consists of two nodes: Try and Catch. The Try node initiates the Try and Catch block and attempts to execute nodes inside of the block. Those nodes are part of the main workflow that you're attempting to execute. Let's call it the main branch. If the execution the Catch Errors node collects the data from the main branch and passes it to the downstream nodes. However, the idea of the Try and Catch block is to continue execution even if there are errors in the main branch. Therefore, the Try and Catch block also has an alternative branch.

If the main branch fails, the Catch Errors node collects the data from the alternative branch. There are sets of Try and Catch Errors nodes with different ports. They can be combined with each other in various ways, depending on the ports needed at the beginning and at the end of the Try and Catch block.

Try and catch, error handeling



Try and catch, error handeling

The way Try/catch work is you can try to execute a workflow and just in case the executed workflow throws an error, then the try catch node allows you to continue your workflow. So instead of having a break in the workflow and the workflow terminates, still have the option to continue the workflow. That's basically what you can achieve with the try catch nodes.

Try and catch, error handeling

And what we are trying to do is we're trying to get the latest sales report, but we do not know whether our colleagues have already well inserted some data in the new sales file. Maybe it exists already, but we do not know whether it works or not. So we currently know that the fourth is empty.

Read with Excel Reader the files Sales4 (the empty table) and Sales5

Under the section workflow control, you will also find error handling nodes and these are mostly the try catch error nodes. So the way they work is you can try to execute a workflow and just in case the executed workflow throws an error, then the try catch node allows you to continue your workflow. So instead of having a break in the workflow and the workflow terminates, you still have the option to continue the workflow.

Try and catch, error handeling

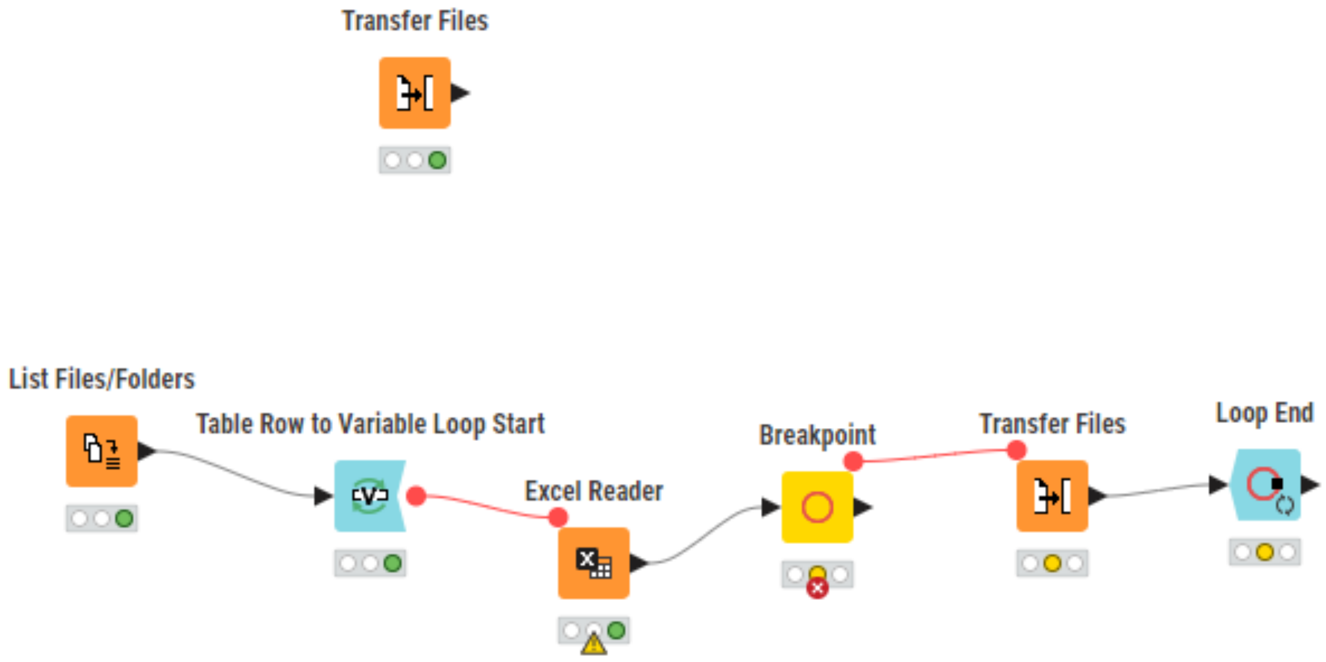
Use three data port here, drag this inside, drag and drop it. And I want to use the fourth. So my newest sales data, which is this one, but only if it actually contains already data. So the way this works with the data ports is the following. We can connect those two and then I would like to have some kind of check whether there is actually data inside.

Insert Try (Data ports) connect with the Sales4

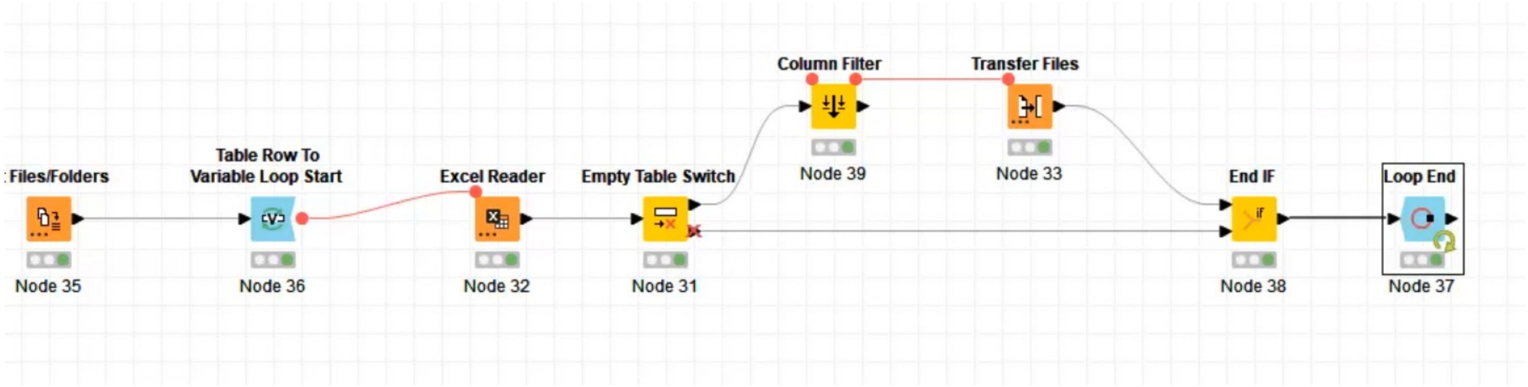
To have some kind of check whether there is actually data inside. Insert a Brakepoint Node and configure it. (empty table)

We use in this case a catch error data port. So we use a data port because we have a data output here so we can connect those two like that. And if you click on the catch errors, you can read through here. But the main point is we have two inputs here. And in case the upper part works, then of course we get here as an output the upper part. If this does not work and throw an error, then we need to give it some additional part in here which then could be used in order to continue the workflow.

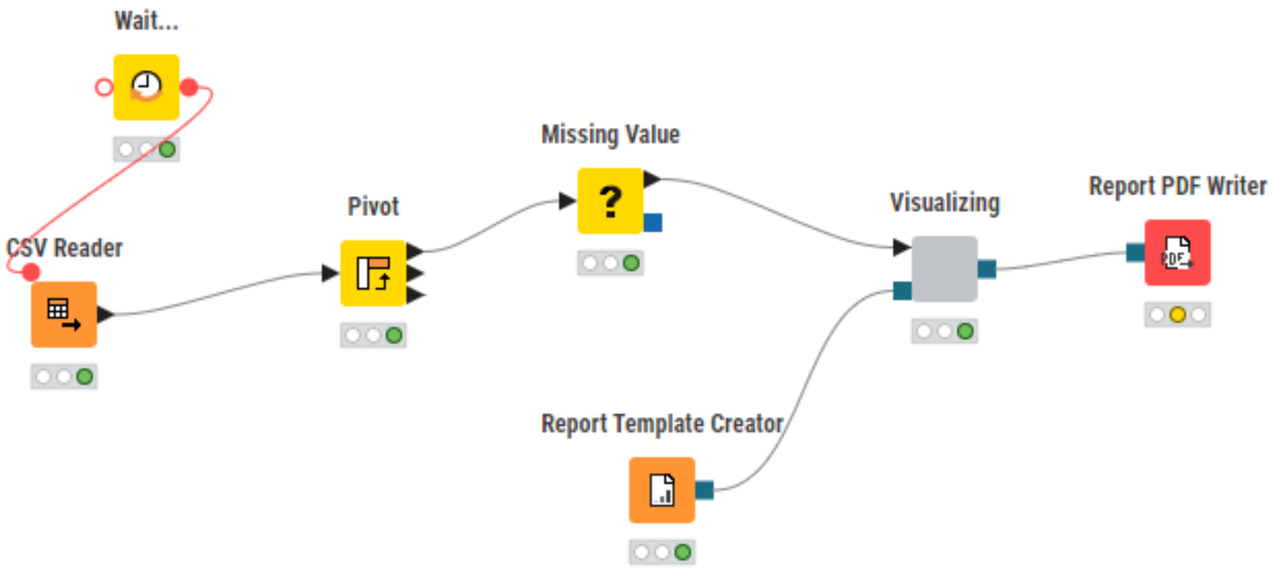
Terminate workflow with loops with breakpoint



Handeling no data



Workflow automatization



Workflow automatization

Explore the workflow controls which are available in Knime.

So in order to do this, of course, we need some data. (Read the transactions2021.csv) So it's a CSV file so I can refer to it using the CSV reader node. Drag this in my workflow, drag and drop, and then right click and go to configure or press F6 on my keyboard works exactly the same. So obviously the default settings for the CSV reader here are fine to me, which means that the column delimiter is a comma that is correctly interpreted by Knime. By default, we get quotes or escape quotes with the quoting in here.

And we got the comment with the hash symbol and very important here has column headers is checked by default.

Workflow automatization

We could adjust a few additional things here in the transformation, advanced settings and so on. And now let's say this is the newest data we have. We receive this file and now we need to quickly generate an odor report. The odor report just means that we got here sales for various countries. Need to know, for each country and each product, what is actually the quantity which has been ordered by customers. To do this, to have it in a pivoting view, we can simply search for pivot.

Grouping Countries, Pivot Product Name,

Workflow automatization

Use the Misson Value Node for change the missing values to 0

Vizualize the results Text View for the title, Table view for the pivot table

Create a component for making a PDF file

Use the Report PDF writer and the Report Template Creator node, and see the results.

Workflow automatization

The node waits for a certain time to a certain or to a certain time or for a file event such as a file creation modification or deletion, and then something gets executed. Now, this is very interesting because we can now connect the wait node to our workflow and then specify when the workflow should be executed. That's meant by automation.

By default, you see that the wait node has two variable ports, so no data ports. Remember, the black ports are the data ports. Red circles, fields or not, fields are variable ports. The difference is just want to repeat this here that the node field circles, these are optionally so you don't need to connect anything to it. The red ones, these are then required to connect to the next node for some output here.

Try the options. Why it"s good? You can create a report:

- Each day in a specified time
- New data comes to our file
- I want to report the newest results (just simply delete it..)

Workflow automatization

It will only run when your Knime and it’s workfolw is opened. If I want to run it as a service, I need PRO license.

Go to the HUB, and upload the workflow to the Pro space

Then create a version history

See the RUN and the Deploy options

Workflow automatization

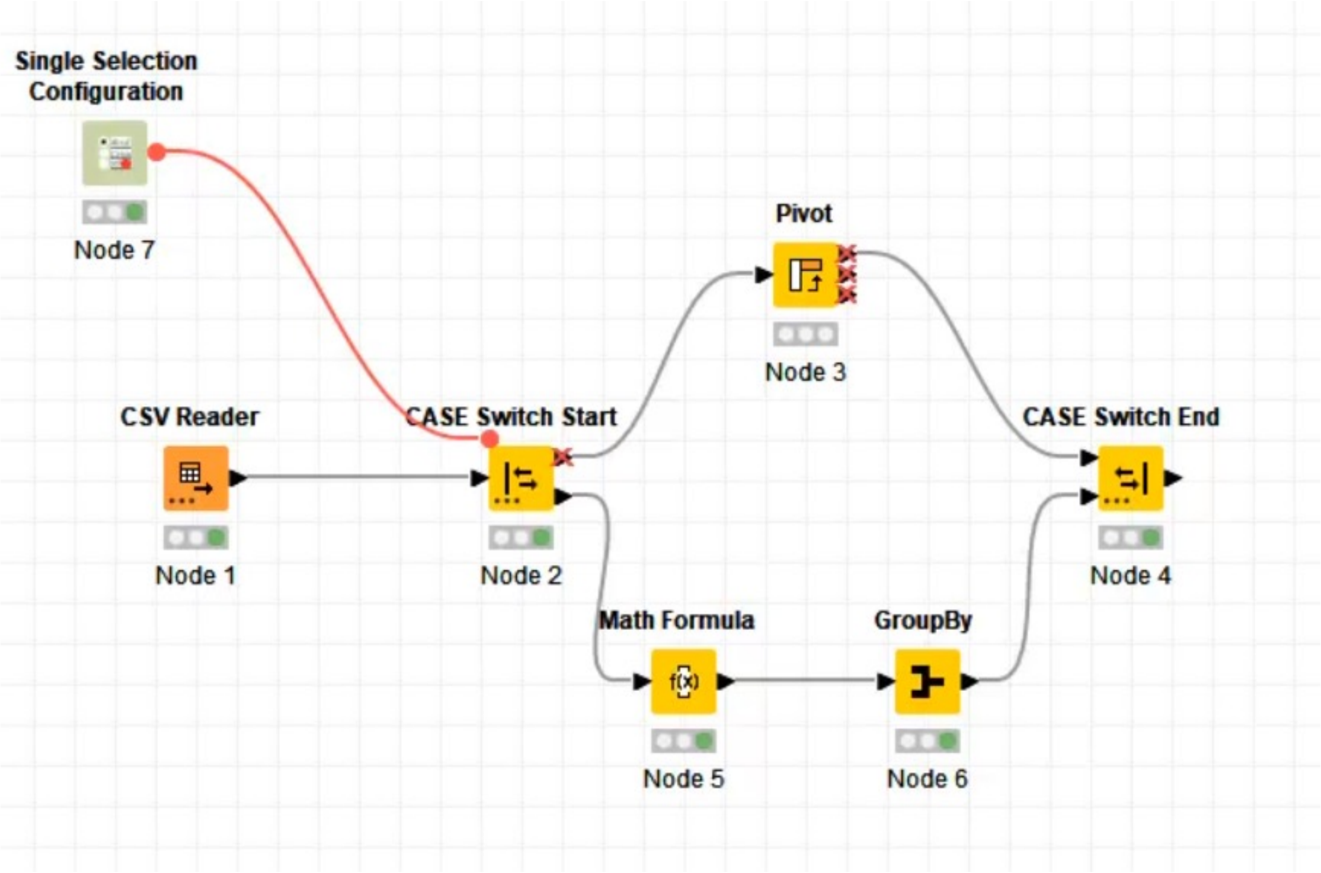
Now the Timer Info node allows you to figure out for individual as well as for the aggregation of nodes, the timing, so the time it takes to execute those nodes, which is very helpful, especially if you have very large, so big complex workflows created and you want to figure out where are actually the bottlenecks.

If you have a very large workflow, which could be the case and it's really complex and you want to figure out, okay, where are actually the bottlenecks?

So maybe one specific node takes a lot of time and then you might think of is there a way maybe to replace the node with another node which might be faster, or can I rebuild my workflow in a way that the workflow works faster?

All these things can be taken into account and this timer info can be very helpful to figure this out.

Case Conditions in KNIME including Flow Variables



Case Conditions in KNIME including Flow Variables

Switch nodes in Knime and learn how to apply them in our knime workflows.

Go to the resource section in here and then I'm going to use this Transactions 2021 file. Now we can of course, go to the node repository and search for our CSV reader node to get the file inside our workflow or if we want to be more efficient.

Check the data, and the data types. At the advanced settings change the decimal separtor!

Case Conditions in KNIME including Flow Variables

To get started with the case statement to check for conditions, we just drag this inside the start.

And then you see by default, it has no ports, which is kind of strange, but this is the new default because all we need to do to add data ports or variable ports or any kind of ports is simply right click on the node and then there's an option to add ports.

The first one is let's say we would like to pivot our data. Search for the pivot node. So that could be one transformation we're going to do. Then you drag pivot inside our workflow, drag and drop it. We connect the upper port to the pivoting node and then we just configure this node.

That gives us then a simple pivot table which tells us how much has been sold for each product in each country.

Case Conditions in KNIME including Flow Variables

- Create a 2nd branch, with Math Formula calculate the Sales (quantity * pirce paid), and with Group by group it by Product names
- Close with a CASE Swith End node
- Search for single Selection Configuration node Let's drag this inside, drop it here, and then let's connect this one to Case Swith Start node.
- Possibe Choices 0 1 (then Pivot or Math formula) connect it with flow variables (single selection index)

Linear regression

Model numerical outcomes using linear regression. One of the most popular approaches in regression analysis. Let’s get started. An overarching goal of regression analysis is to model numerical outcomes based on available input features. In linear regression, you model the target variable y as a linear combination of input features x . With one input feature, the model describes a regression line.

The slope and intercept of the line are controlled by the regression coefficients a1 and a0, respectively. Fitting a linear regression model means adjusting these coefficients to best describe the relationship between x and y To find the best regression coefficients, you calculate the error between the observed data and the linear regression model at each data point, referred to as residuals. The best model minimizes the residuals for all data points simultaneously.

The linear regression learner node in KNIME can estimate this model. The learner node takes a dataset at the input port and produces the regression model at the output port. In the configuration window, specify the target column (outcome variable) and input features.

Linear regression

We will use a car dataset to predict miles per gallon (MPG) based on the numerical feature horsepower (HP). After training the model, in the output table, you can examine the regression coefficients, their standard error, t-statistic value, and p-value. The model trained in the learner node can be used in the regression predictor node.

This node takes the trained model and a dataset, and produces predicted outcomes in the output data set. The performance of the model can be summarized by various goodness-of-fit metrics calculated in a numeric scorer node. In the configuration window, you specify the target as the reference column, and the predicted outcome as the predicted column.

In this example, your model had an R-squared value of 0.588, indicating that 59% of the variability was explained by this linear regression model.

In this specific case, the model explains more than half of the variability in the dependent variable, which is often considered a useful outcome, especially in social science or economic problems where there is a lot of noise. The remaining 41.2% remains unexplained; this could be due to measurement error, omitted explanatory variables, or non-linear relationships. Therefore, an R2=0.588 R 2

=0.588 is not perfect, but it is already a result that can be evaluated as meaningful for most practical applications.

Training and Applying Decision Trees

Apply a decision tree in the KNIME analytics platform! Decision trees are powerful tools for solving classification problems.

A decision tree is like a flowchart or a series of yes/no questions that help make a decision or prediction. It starts from a main question called the root node, then branches out into further questions (called decision nodes) based on the answers. This splitting keeps going until it reaches an end point called a leaf node, which gives the final answer or prediction.

How it Works Step-by-Step:

1. Start with one big group of data (the root).
2. Ask the best question that divides this group into smaller groups.
3. For each smaller group, ask another question to split it further.
4. Continue splitting until every group is simple enough to assign a clear result (leaf).
5. Use the tree by following the questions to reach a leaf that gives the prediction.

Training and Applying Decision Trees

This example covers how to train a decision tree on a wine dataset to classify wines as either red or white.

First, let’s explore our dataset. Open the output of the table reader node. The dataset includes multiple numerical columns representing the chemical components in wine and one nominal column for the wine quality. These columns are our input features.

The last column, indicating wine color is our target variable. Next, add the Decision Tree Learner Node from the node repository to our workflow and connect it with our training data.

This node takes data from the training set and produces a model, indicated by the blue squares. Double-Click the Decision Tree Learner Node to open its configuration window.

Training and Applying Decision Trees

First, set the class column to “Color” The decision tree algorithm will split the data based on the best input feature at each iteration.

You can choose between different metrics for splitting, such as Gini index and gain ratio. Numeric attributes are always split into two subsets, while nominal attributes can create multiple branches unless you enforce binary splits.

For nominal columns with too many values,   you can set a maximum number for binary splits to avoid excessive computation. If nominal columns lack domain information,   enable the checkbox to skip these columns for faster calculation.

Large trees risk overfitting, while small trees may not learn enough. You can control tree size by setting a minimum number of records per node, which stops further splitting if this threshold is not met.

To avoid overfitting, you can prune the tree. KNIME offers the Minimum Description Length (MDL) and reduced error pruning methods. Reduced error pruning, activated by default, replaces each node with its most popular class if accuracy doesn't decrease.

Configure the number of records to store for view and the number of threads to optimize node execution speed without affecting the model.

Once configured, execute the Decision Tree Learner node and inspect the output using the Decision Tree View to see how the tree classifies the wine data.

Class and Overall Accuracy Statistics

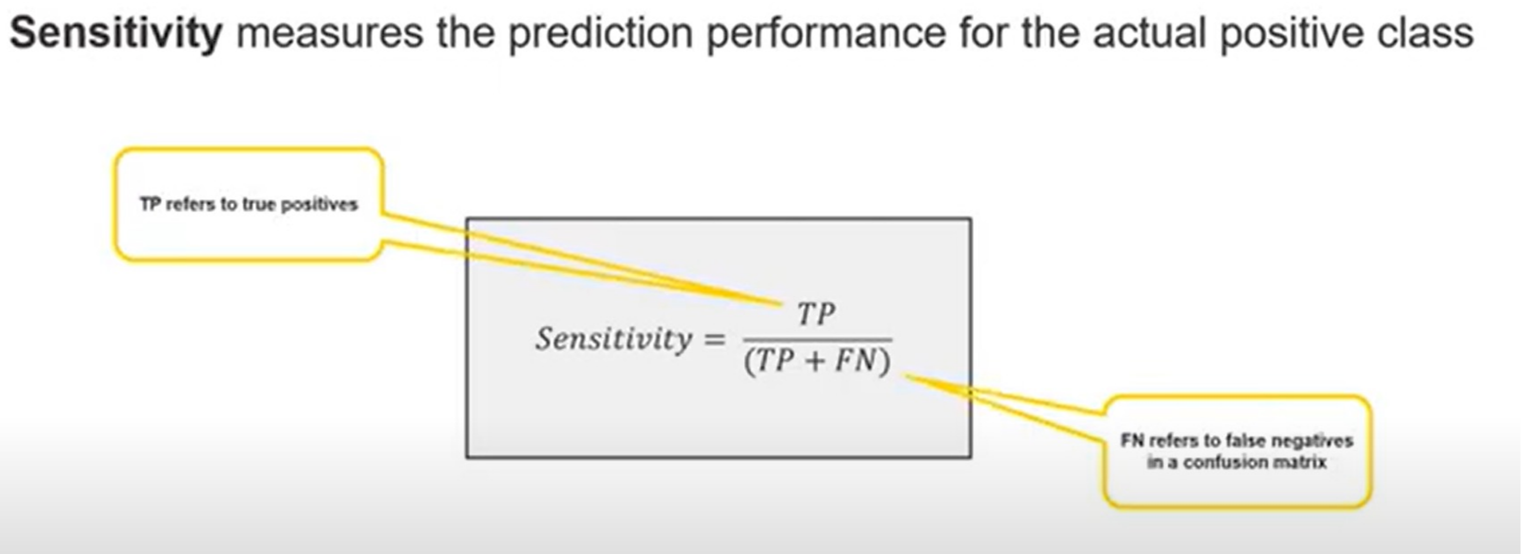
https://hub.knime.com/knime/spaces/Education/Educational%20Videos%20Demo%20Workflows/Class%20and%20Overall%20Accuracy%20Statistics~hclq-qMLxPODTEYT/current-state

Take a look at class and overall accuracy statistics that report the performance of a classification model. These statistics are calculated based on the counts in the confusion matrix. Here you see an example classification model that predicts customer satisfaction for a fictitious Airline using the Logistic Regression algorithm.

The predictor columns contain passenger and flight information, such as: Customer Type, Type of Travel, Flight Distance, and Seat Comfort. The target column is called satisfaction and has two classes: **satisfied and dissatisfied**. The Scorer (JavaScript) node reports the performance of the classification model in its interactive view.

The class and overall accuracy statistics reporting its performance.

Class and Overall Accuracy Statistics

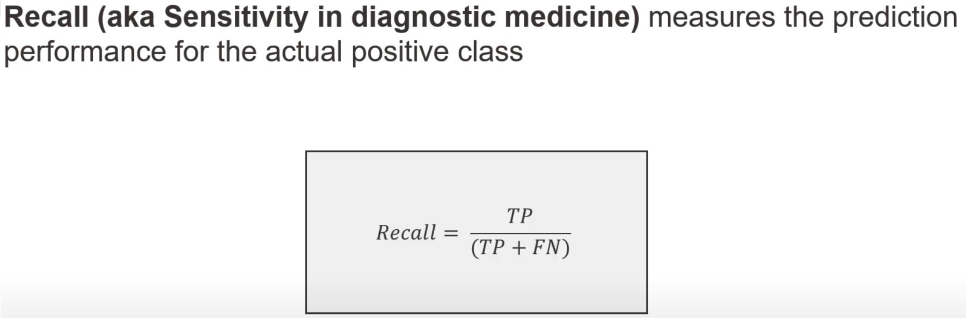


Sensitivity measures the prediction performance for the actual positive class

TP refers to true positives

$$Sensitivity = \frac{TP}{(TP + FN)}$$

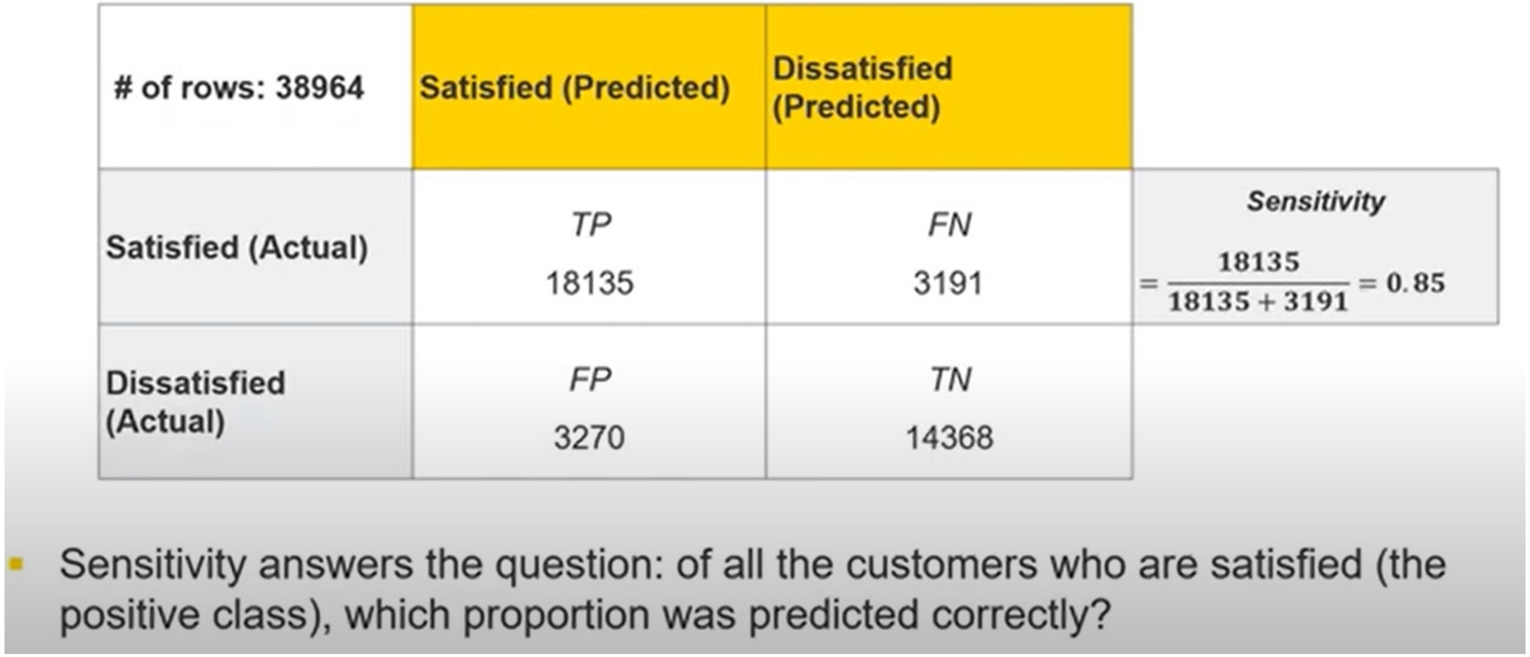
FN refers to false negatives in a confusion matrix



Recall (aka Sensitivity in diagnostic medicine) measures the prediction performance for the actual positive class

$$Recall = \frac{TP}{(TP + FN)}$$

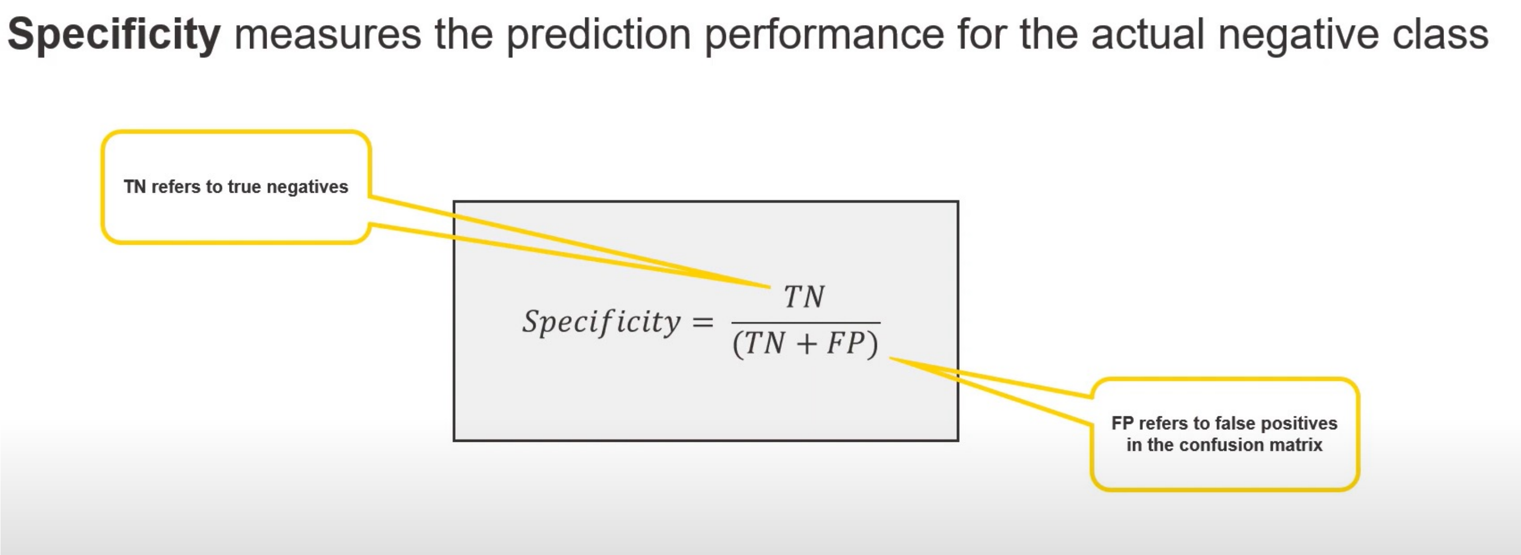
Class and Overall Accuracy Statistics



# of rows: 38964	Satisfied (Predicted)	Dissatisfied (Predicted)	
Satisfied (Actual)	*TP* 18135	*FN* 3191	***Sensitivity*** $= \frac{18135}{18135 + 3191} = 0.85$
Dissatisfied (Actual)	*FP* 3270	*TN* 14368	

- Sensitivity answers the question: of all the customers who are satisfied (the positive class), which proportion was predicted correctly?

Class and Overall Accuracy Statistics



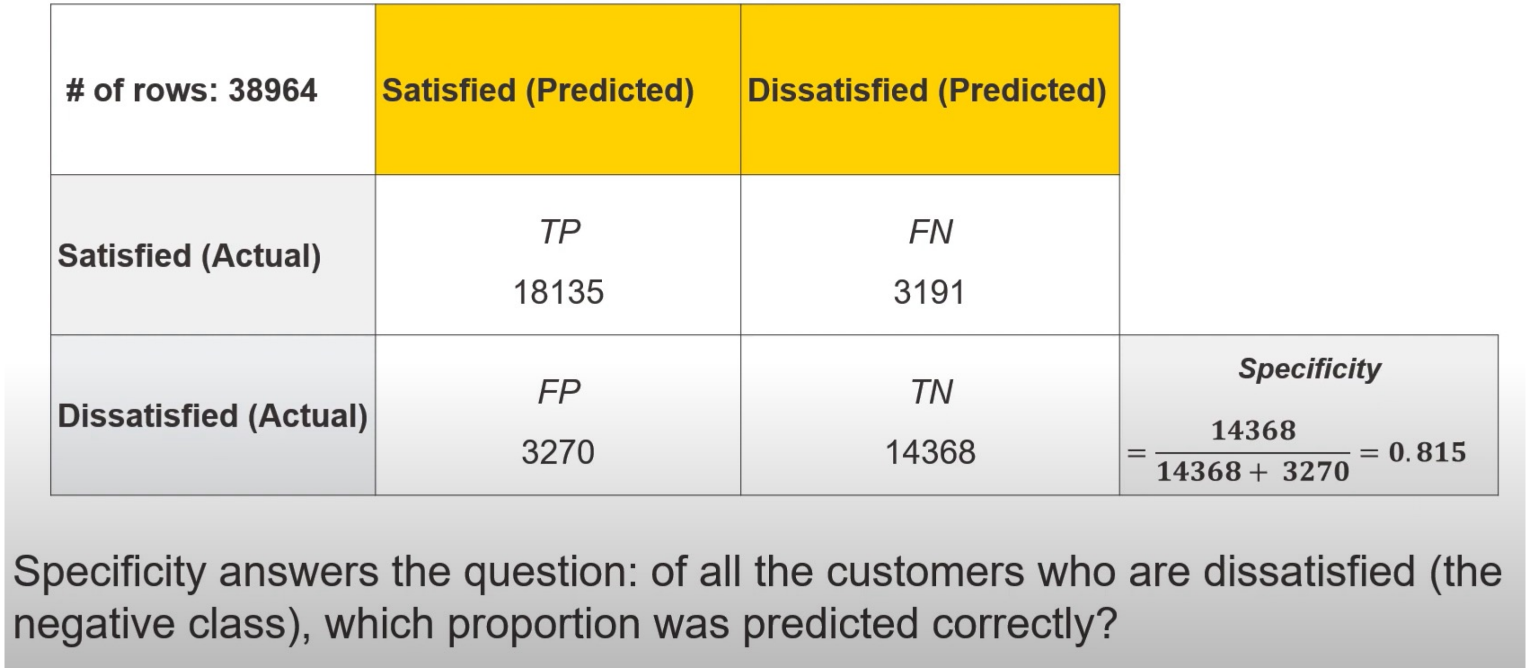
Specificity measures the prediction performance for the actual negative class

TN refers to true negatives

$$Specificity = \frac{TN}{(TN + FP)}$$

FP refers to false positives in the confusion matrix

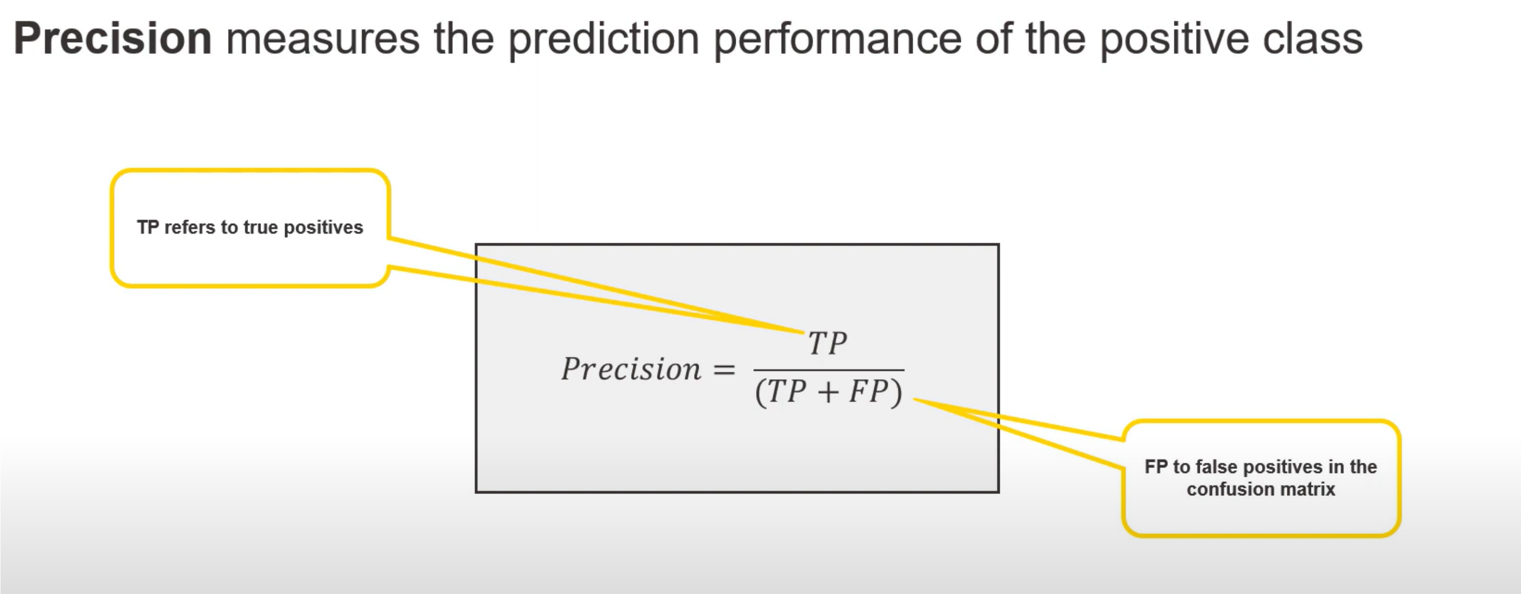
Class and Overall Accuracy Statistics



# of rows: 38964	Satisfied (Predicted)	Dissatisfied (Predicted)	
Satisfied (Actual)	*TP* 18135	*FN* 3191	
Dissatisfied (Actual)	*FP* 3270	*TN* 14368	*Specificity* $= \frac{14368}{14368 + 3270} = 0.815$

Specificity answers the question: of all the customers who are dissatisfied (the negative class), which proportion was predicted correctly?

Class and Overall Accuracy Statistics



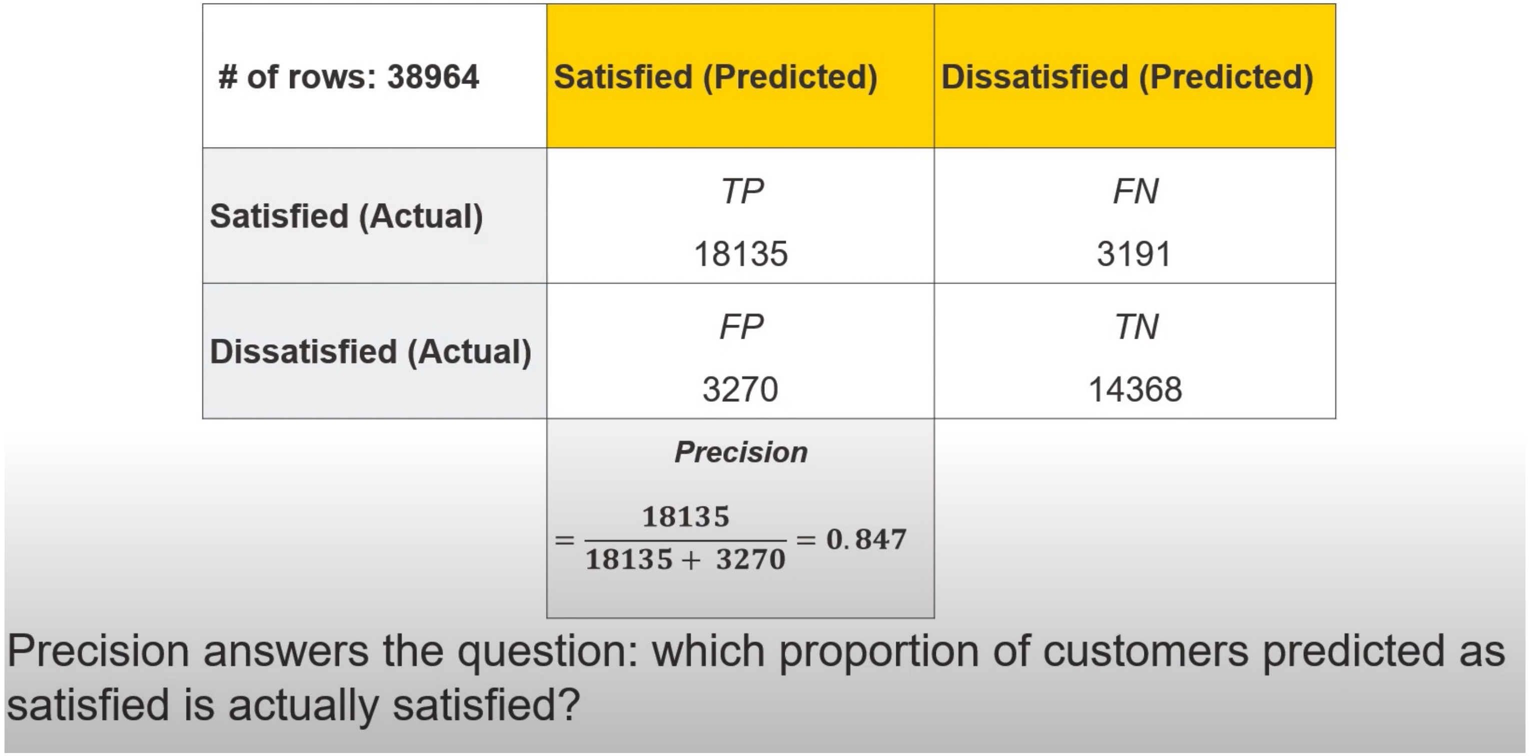
Precision measures the prediction performance of the positive class

$$Precision = \frac{TP}{(TP + FP)}$$

TP refers to true positives

FP to false positives in the confusion matrix

Class and Overall Accuracy Statistics



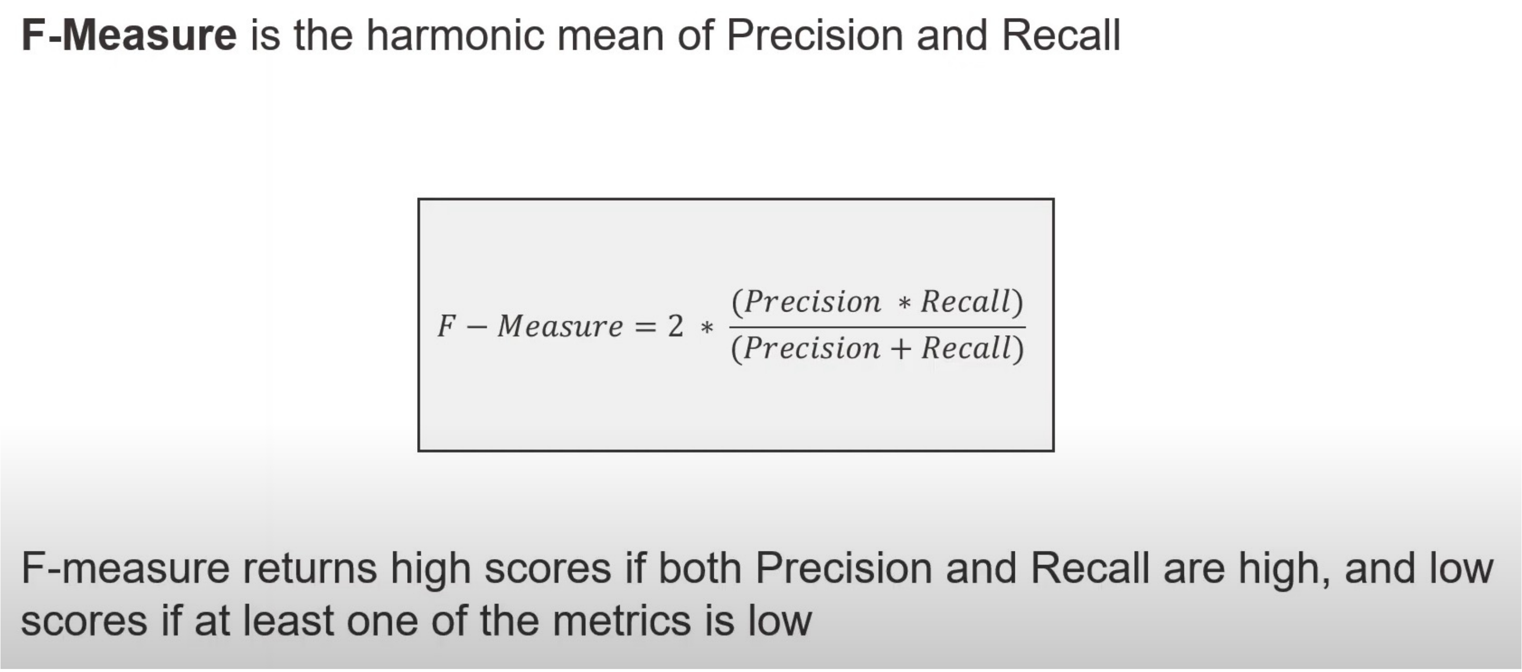
# of rows: 38964	Satisfied (Predicted)	Dissatisfied (Predicted)
Satisfied (Actual)	*TP* 18135	*FN* 3191
Dissatisfied (Actual)	*FP* 3270	*TN* 14368

Precision

$$= \frac{18135}{18135 + 3270} = 0.847$$

Precision answers the question: which proportion of customers predicted as satisfied is actually satisfied?

Class and Overall Accuracy Statistics



F-Measure is the harmonic mean of Precision and Recall

$$F - Measure = 2 * \frac{(Precision \ * Recall)}{(Precision + Recall)}$$

F-measure returns high scores if both Precision and Recall are high, and low scores if at least one of the metrics is low

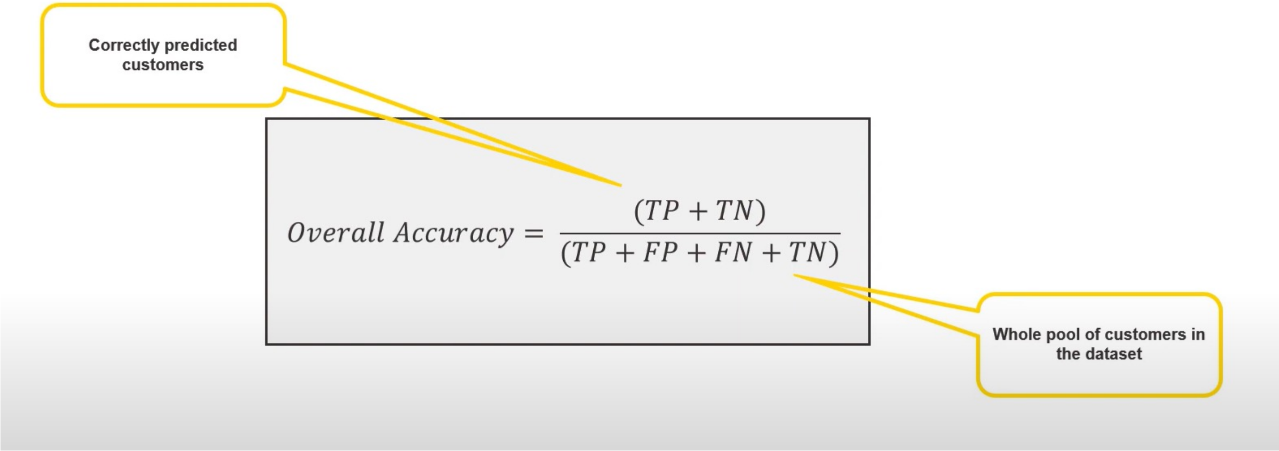
Class and Overall Accuracy Statistics

Overall accuracy statistics report the overall model performance in one metric

To calculate them we need all 4 counts in the confusion matrix

Let’s have a look now at these overall accuracy statistics:

- Overall Accuracy
- Overall Error
- Cohen’s kappa

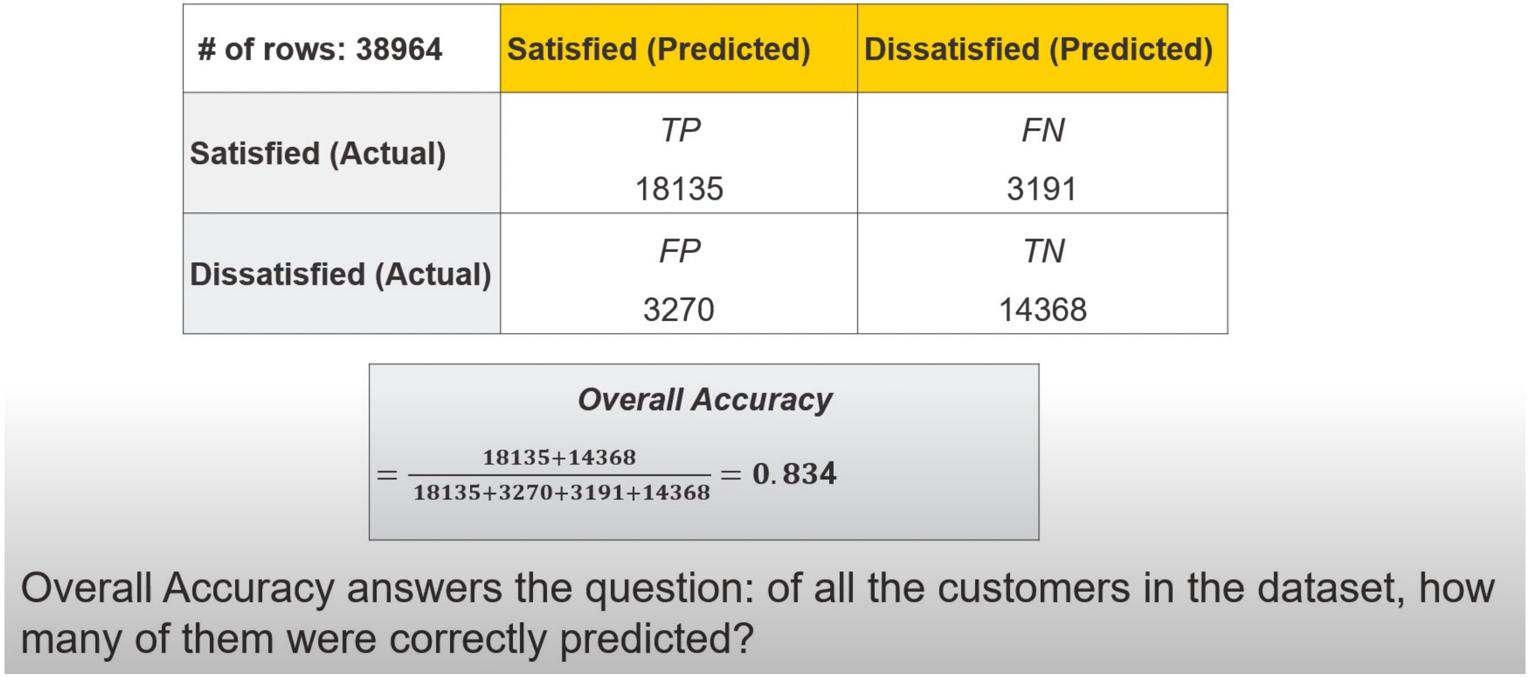


Correctly predicted customers

$$Overall\ Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)}$$

Whole pool of customers in the dataset

Class and Overall Accuracy Statistics



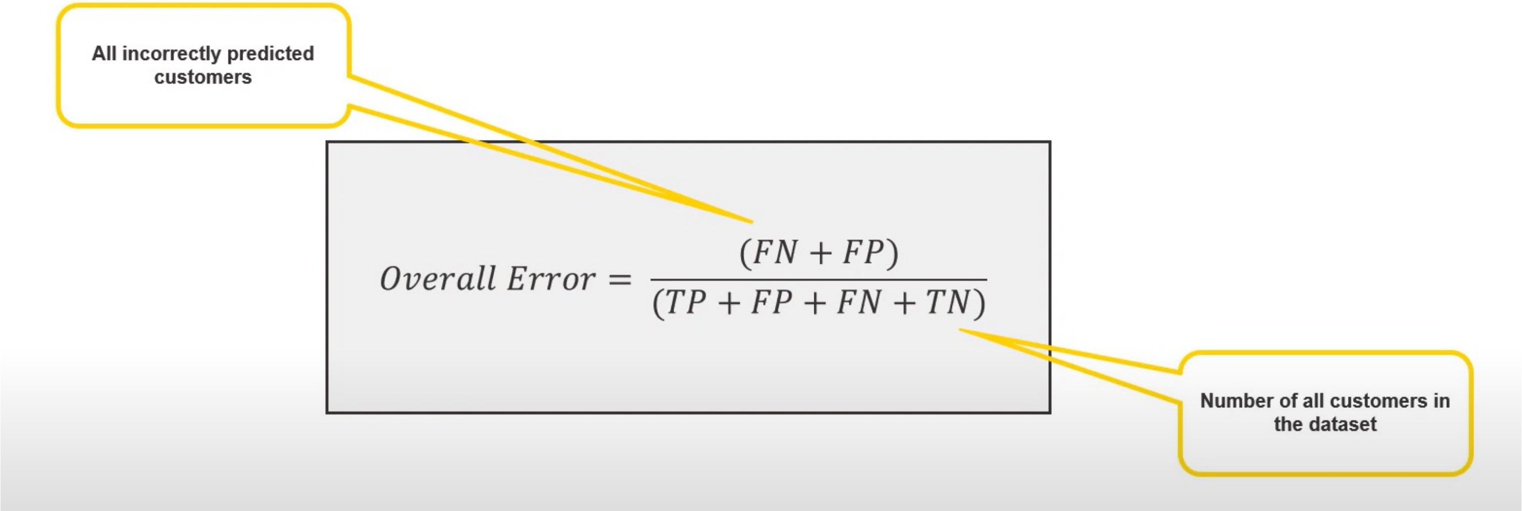
# of rows: 38964	Satisfied (Predicted)	Dissatisfied (Predicted)
Satisfied (Actual)	*TP* 18135	*FN* 3191
Dissatisfied (Actual)	*FP* 3270	*TN* 14368

Overall Accuracy

$$= \frac{18135+14368}{18135+3270+3191+14368} = \mathbf{0.834}$$

Overall Accuracy answers the question: of all the customers in the dataset, how many of them were correctly predicted?

Class and Overall Accuracy Statistics

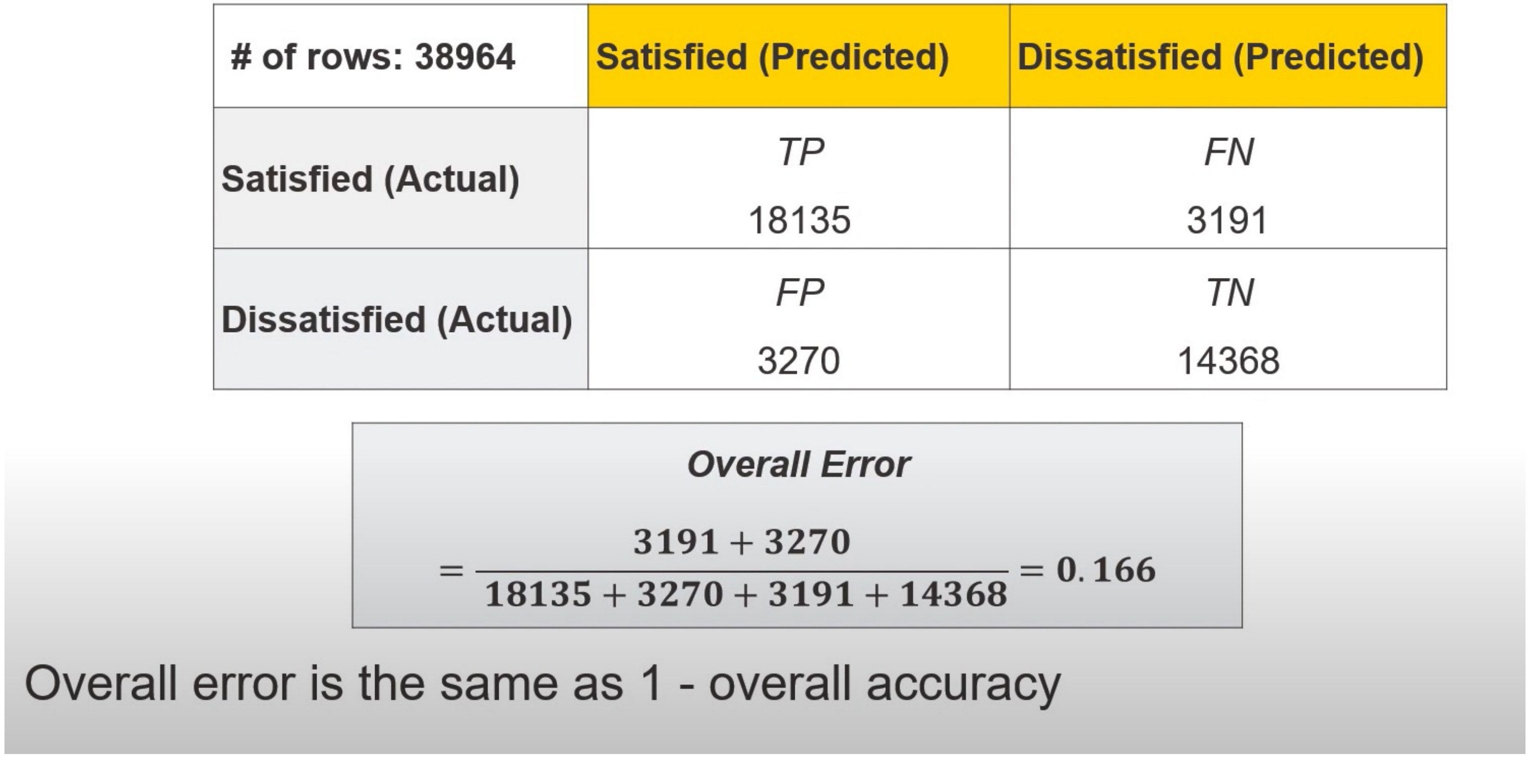


All incorrectly predicted customers

$$Overall\ Error = \frac{(FN + FP)}{(TP + FP + FN + TN)}$$

Number of all customers in the dataset

Class and Overall Accuracy Statistics



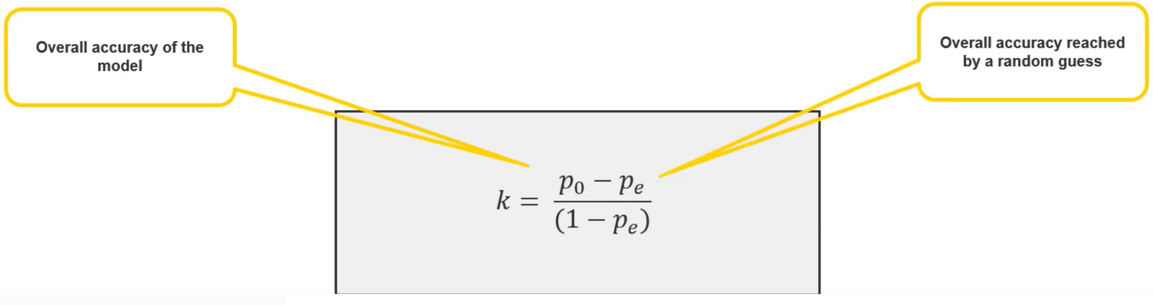
# of rows: 38964	**Satisfied (Predicted)**	**Dissatisfied (Predicted)**
Satisfied (Actual)	*TP* 18135	*FN* 3191
Dissatisfied (Actual)	*FP* 3270	*TN* 14368

Overall Error

$$= \frac{\mathbf{3191 + 3270}}{\mathbf{18135 + 3270 + 3191 + 14368}} = \mathbf{0.166}$$

Overall error is the same as 1 - overall accuracy

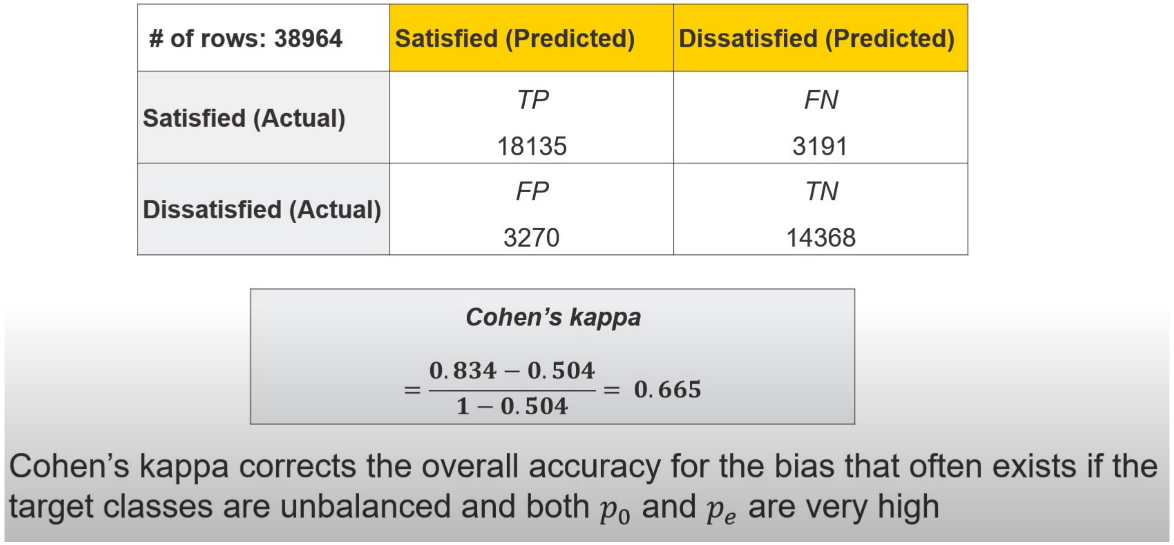
Class and Overall Accuracy Statistics



Overall accuracy of the model

Overall accuracy reached by a random guess

$$k = \frac{p_0 - p_e}{(1 - p_e)}$$



# of rows: 38964	Satisfied (Predicted)	Dissatisfied (Predicted)
Satisfied (Actual)	*TP* 18135	*FN* 3191
Dissatisfied (Actual)	*FP* 3270	*TN* 14368

Cohen's kappa

$$= \frac{0.834 - 0.504}{1 - 0.504} = 0.665$$

Cohen's kappa corrects the overall accuracy for the bias that often exists if the target classes are unbalanced and both p_0 and p_e are very high

Improving the Cohens Kappa

https://hub.knime.com/knime/spaces/Education/Educational%20Videos%20Demo%20Workflows/Cohens%20Kappa~unY5HpYeHjG8BnpQ/current-state

This workflow demonstrates how Cohen's kappa can be used to evaluate the performance of a classification model when dealing with imbalanced data. We also show how Cohen's kappa obtains greater values not only due to better model performance, but also because of a more balanced target class distribution.

Improving the Cohens Kappa

This node (SMOTE) oversamples the input data (i.e. adds artificial rows) to enrich the training data. The applied technique is called SMOTE (Synthetic Minority Over-sampling Technique) by Chawla et al.

Some supervised learning algorithms (such as decision trees and neural nets) require an equal class distribution to generalize well, i.e. to get good classification performance. In case of unbalanced input data, for instance there are only few objects of the "active" but many of the "inactive" class, this node adjusts the class distribution by adding artificial rows (in the example by adding rows for the "active" class).

The algorithm works roughly as follows: It creates synthetic rows by extrapolating between a real object of a given class (in the above example "active") and one of its nearest neighbors (of the same class). It then picks a point along the line between these two objects and determines the attributes (cell values) of the new object based on this randomly chosen point

Parameter Optimization Loop

https://hub.knime.com/knime/spaces/Examples/04_Analytics/11_Optimization/06_Parameter_Optimization_two_examples~lkw5Tu3h_pVXzVUe/most-recent

The goal of using a parameter optimization loop is to find   the best set of parameters for your prediction model. Like all other loops, the parameter optimization loop follows the classic KNIME loop motif, which includes a loop body located between the loop start node and the loop end node.

Parameter Optimization Loop

To perform parameter optimization, you use the Parameter Optimization Loop Start and Parameter Optimization Loop End nodes. These two nodes surround the loop body where you train the prediction model.

The Parameter Optimization Loop Start node iterates through a list of parameter sets, applying a new set of parameters to the learner node at each iteration. The Parameter Optimization Loop End node collects these iteration results and compares them to previous ones.

Start with a pre-prepared workflow that reads a dataset, splits it into a training set and a test set using the Partitioning node, trains the classification model with the Decison Tree Learner node, applies the model to the test dataset using the Decison Tree Predictor node, and uses a Scorer node to calculate accuracy and evaluate the model's performance. Our goal is to find the set of parameters for Decison Tree Learner algorithm that leads to the highest accuracy on the test set.

Parameter Optimization Loop

First, look at the configuration window of the Decision Tree Learner node to identify which settings can be optimized.In this example, we will optimize the "Minimum Number of Records per Node" setting. The aim is to find the optimal value for this setting that leads to the highest accuracy.

This node has only one output port of type flow variable. In the configuration window, you can add a parameter by clicking the "Add New Parameter" button. Define the parameter name, start value, stop value, and step size. For example, you can create a new parameter named "Min Number of Records" with a start value of 2, a stop value of 15, and a step size of 1.

You can add more parameters if needed. You can choose between two search strategies: Brute Force or Hill Climbing. The Brute Force strategy checks all possible parameter combinations and returns the best one, meaning it will train a model for each value between 2 and 15 in our example.

The Hill Climbing strategy requires less computational effort as it starts with a random set and proceeds with only the direct neighbor values according to the given intervals and step sizes. The best value combination among all neighbors becomes the starting point for the next iteration. If no neighbor improves the accuracy, the loop terminates.

After executing the start node, you see that you get one flow variable with our defined parameter name and the start value of 2.

Parameter Optimization Loop

When we execute the Parameter Optimization Loop End node, the whole loop is executed. With the Brute Force strategy selected, the model is trained for each parameter value between 2 and 15.

Once the loop execution is complete, we can examine the resulting tables. The table at the first output port gives us the parameter set corresponding to the highest accuracy, while the second output table provides the accuracy for all tested parameters.

Case Study 1. Energy drink transactions in 2022

- Download the files : https://drive.google.com/drive/folders/1KZC-vGg52S2ySJpG6YfEQ2Pnb_TZonLS?usp=drive_link
- You find the data in this section.
- It makes sense to have a look at it just to better understand what kind of data we're dealing with.
- Open Transactions Apr22.csv and take a look at it. So this is an example is one of the transaction files. We have one transaction for each month. We have 12 different kinds of transaction files
- Then open Sales Rep.xlsx and Product Info.xlsx
- (a comma as my decimal separatorn here, we need to deal with because nine by default only takes a dot as a decimal separator.)
- Open RegionDictionary.xlsx

Case Study 1.2

- Consolidate the transactions data into one giant table
- Explore Import options in Knime
- Import data sources Cleaning the Sales Area in the Sales Rep Sheet
- Saving our files to disk
- Enrich our data with currency information using a web api
- Joining data – start combining our results (additional cleaning involved)

Case Study 1.3

- Create a new workflow
- Import an Excel file (Product_Info.xlsx) with Excel Reader Node
- We can see a preview of the data, each of the columns in here we also can see the data type.
- Investigate the options (Data area, Advanced, Transformation)
- (Files and folder only works if the files have the same structure.)
- (Red question mark means it's an empty cell)
- Name the node (Product Information)

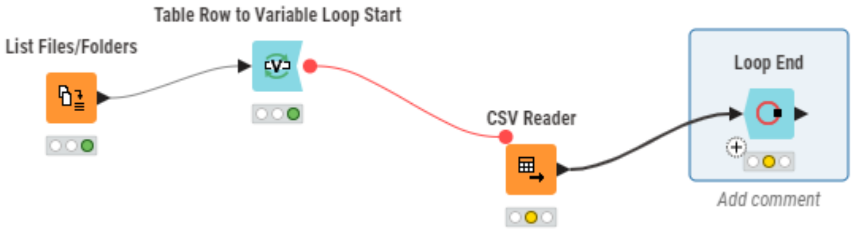
Case Study 1.4

- Merge transaction files together into one giant table
- Drag and drop the files to the canvas (workflow editor).
- Merge with concenate node (There are several different port types, and by default you should just keep in mind that normally only the same port types can be connected together.)
    - Union / Intersection
    - Create new input ports
- Merge with Flies in Folder option
- Merge with loop

Case Study 1.5

Merge with loop conception (iteration)

- read multiple files file by file into Knime
- a quite powerful concept which is used in many advanced workflows
- List Files/Folders node (It lists all the files or folders inside one specific folder in the folder path
- Table Row to Variable Loop Start node (red circle - is a variable connection - path variable)
- Use a CSV Reader node (flow variables – file selection – path (Path form the Loop start node
- Loop End node



Case Study 1.6

Cleaning the Sales Area in the Sales Rep Sheet

- Read the Sales Rep.xlsx
- clean the sales area in the sales rep sheet and by cleaning, replace actually the abbreviation with the correct names,
- Read RegionDictionary.xlsx
- Excel related solution is a lookup, with database concept it is a join.
  - Value lookup node (read the discription)
  - Replace the Sales Area column
- Save the files to disk
  - Excel Writer
  - CSV Writer

Case Study 1.7

Add currency informaion using the web

- we have received our transactions in a certain kind of currency and now we want to convert it. For this conversion we are interested in the exchange rate, we would like to have the latest exchange rate from the web.

- There are various options which are available, various service providers. We use European Central Bank free service because we don't need here registration.

- GET Request node use the URL: https://www.ecb.europa.eu/stats/eurofxref/eurofxref-daily.xml (Status 200 - Request was succesful, 404 – failed)

- Get this now in a proper format in time. And there are several ways to do it. But this is currently XML. XML to JSON node (Append a new column) See the two formats!

- Convert it into JSON first and then use the JSON Path node to extract the data (A JSON file contains JavaScript Object Notation, a lightweight, human-readable text format for exchanging data between systems.)

- JSON Path allows us to extract JSON data (use the JSON for input)

- Drill down exactly to get the currency in this case US dollar, simply click on itand then the Rate – Add single query)
- Change the name with doube click, (Currency symbol - string, USD Exchange rate – double)
- Filter with coloumn filter and write to Excel file (if it exsists owerwirite it)

Case Study 1.8

Clean the Product info data (country column only the first entry is filled – missing value)

- Ask the K-AI! Write a promt! Example: *„There are missing values in a column in the selected Excel node. Can I fill in the missing values with a previous value?”*
- Missing Vaule node

Combining the cleaned data Sales_Rep_cleaned, Transactions_ALL

- Map the Sales Manager with the transactions
  - In Sales_Rep-cleaned each product name has a region and for each combination of region and product name we have one sales manager
  - In Transactions the product name and the region is also available.
- Joiner Node
  - Two citeria: Map Sales Area – Regions AND Product name – Product name (Try INNER JOIN, LEFT OUTER, FULL OUTER) Split join result into multiple tables, see the differentes
  - Remove the duplicated columns

Case Study 1.9

- Inner Join (Only rows with matching key values in both tables appear in the result. If a key is missing from either table, that row is excluded.
- Left Outer Join (All rows from the first (left) table are kept. If there's no match in the second (right) table, the missing values are set as null (or "missing").
- Right Outer Join (All rows from the second (right) table are kept. If there's no match in the first (left) table, the missing values are set as null.
- Full Outer Join (All rows from both tables are included. If a row has no match in the other table, the missing side will have null ("missing") values.

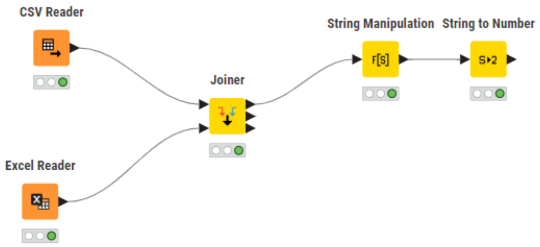
These join types determine how data from two tables is merged, based on whether matching keys are required on both, either, or both sides.

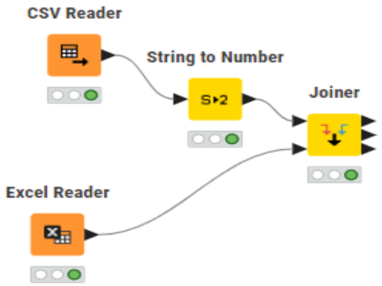
Case Study 1.10

See the results of INNER JOIN and the data types

Price paid is a String, because of the decimal saparator! (coma vs dot)

- Solution 1: In the CSV Reader Node, Advances Settings, Change the Decimal separator -> Warning
- Solution 2. : K-AI („how i can change in each row of a column a decimal separator from coma to dot?” String Manipulation Node and String to Number Node
- Solution 3.: String to Number Node between the CSV Reader and Joiner (set decimal separator ” , ”





Case Study 1.11

Join the Product_Informaion_cleaned and the joined table

- matching columns – Country, Product name
- need to join the data, but the Joiner node has only 2 input ports, use another Joiner Node
- Save it to a csv file Transactions2.csv (Csv Writer)

Remove duplicate entries (if any)

- Use the Duplicate Row Filter (All columns included)

Remove Quantities below or equal to zero

- Use the Row Filter Node

Case Study 1.12

Remove Rows with Product Name #NV

- Use the Row Filter Node, because the OR relation

Calculate the Profit and the Revenue

- Math Formula for Revenue (Quantity * Price paid)
- the Math Formula node allows us to do any kind of mathematical operations, in this case (Price Paid – Productions Costs) * Quantity

Join the currency exchange rate data, which we extract from the web

- get the exchange rate in our workflow
- need to somehow now map these two different kinds of data sources (no column no column which we can use in order to map these two different kinds of data sources) – need an USD row
- Constant Value Column (append Currency Symbol – String – USD)
- Join the nodes with Joiner

Case Study 1.13

Calculate the profit in euro

- Use the Math Formula node

Save the data - Transactions3.csv

Tableau and Power BI extension Install

- Tableau wirter (convert to .hyper file)

There's only the option here to send the data directly to the power BI service.

- can be done via another node which is the Microsoft authentication node and Send to Power BI
- See it in your workspace and do the magic (Üzleti összefüggések felftárása – Üres jelentés) (https://app.powerbi.com/home?experience=power-bi)

Case Study 1.14

Vizualize the data in Power BI

- power BI for this example, simply because we can all use power BI desktop for free, so you can download it and use it
- The problem for me specifically is for instance in Hungary, the comma is the decimal separator.
  - 1. soluiton: convert the dot to coma
  - 2. Power Query – Change date type – Using locale (Data type – Fix Decimal Number, Locale – English – US)
- Vizualize in Map the county by profit
- Protuct and Quantity by dounat chart

Case Study 1.15

Data Visualization in KNIME

- Views: Bar chart (Category product name – Sum – Profit)
- Pie chart – Category – Country, Aggregation Sum – Frequency Qunatity, Treshold - Label value format - Propotion
- prefer them by tools regarding power BI, Tableau,

Sort the plotting of bars in Knime

- In a Bar Chart Node it is not implemented yet
- Use the Sorter Node

Report in Knime

To create simple and advanced reports, we have to install The KNIME Reporting Extension, that allows you to create static reports based on the results of your workflows. You can automatically generate and distribute customized reports for recurring events such as month-end close, quarterly performance, or on-demand statistics. For instance, you can send a PDF email report based on your data.

- Step 1: Add views to your workflow and create a component
  - Add rich text to your report. Search for the Text View node and then, drag it to your workflow canvas. Right click it to open the configuration dialog. It shows a preview of the text. You can manipulate it in the Rich Text Content editor on the right. At the top of the editor you can choose between the following formatting options. („Sales report 2022”)
  - Create a component. Fundamentally, the report you are creating consists of a component's composite view. To include the view nodes in your report, wrap them in a component. First select the nodes, then click the Create component button in the toolbar at the top. Open the component by double-clicking it while holding down the Ctrl key, or right-clicking and choosing Component → Open component. The sub-workflow contained by the component is displayed.

Report in Knime

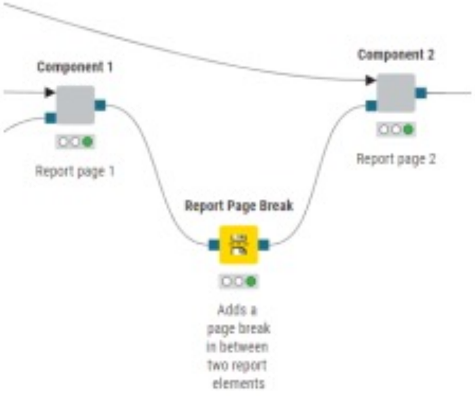
Open the layout editor. To customize the composite view, use the layout editor. Select Open layout editor from the toolbar at the top. The layout editor will automatically create a layout, but it also allows you to customize it with a drag and drop grid. If you want to resort the order of the view elements, you can simply drag and drop them to the desired position.

Enable the reporting function. To enable the component to output the views for a report, tick the Enable Reporting checkbox at the bottom of the layout editor and click Finish. This creates a Report input and a Report output port to your component. **They will be visible as blue squares adjacent to the component once you exit.**

Step 3: Customize the template of your report. Before you can connect your component views to a reporting node, you need to customize the page size and orientation using the Report Template Creator node. Add it to your workflow and connect its output port to the existing Report input port on the left side of the component**.**

Report in Knime

If you want to add a second page to your workflow, use the Report Page Break node. It prevents your views from being cut off at the end of a page in your report file. You can find the Report Page Break node in the node repository. Add it in between two components to separate their contents with a page break on the report



Report in Knime

Step 4: Write to file In order to share the previously created content as a document, you need to save it to a file first. From the Report output port of the component creating the report, you can drag a connection and select compatible writer nodes. You can save your report as either a PDF or HTML file. (Make sure that the Enable Reporting checkbox at the bottom of the component’s layout editor is selected and that a Report Template Creator node is connected to your component, as described in the previous steps. Otherwise, you cannot write the report to a file.)

To save your report as a .pdf file, add the PDF Writer node to your workflow. This node allows you to write your report to a PDF file in a specified output location.

Generative AI in Knime

- How Does KNIME K-AI Work?
- Integrated with KNIME Analytics Platform and Community Hub — accessible to registered users.
- Works in two modes:
  - Q&A mode: Responds to user queries about workflow design or data science concepts.
  - Build mode: Helps construct workflows by recommending and adding components based on user inputs.
- Collaborates interactively with the user to refine workflows.
- BUT only works with SELECTED and EXECUTED nodes!!

Example:

„Filter the data and keep only the columns Country, Quantity and Profit! Then keep only the top 10 rows" – Transactions3.csv

User types: "Add a node to handle missing data."

K-AI responds with suggested nodes like "Missing Value" or "Imputation" and can add the selected node directly into the workflow.

Generative AI in Knime

What is KNIME K-AI and What Is It For?

- KNIME K-AI is an AI-powered assistant integrated into KNIME for building and optimizing data workflows.
- It helps users create workflows faster by suggesting nodes, connections, and automating repetitive tasks.
- K-AI can answer technical questions and provide guidance on data preprocessing, modeling, and analytics.

Example:

- „How do is retrieve web data to Knime and load it as a table"
  - Webpage retriever – Xpath nodes (try to solve the earlier task)
- User wants to build a customer churn prediction workflow.
  - K-AI suggests relevant nodes like "Data Preprocessing," "Random Forest," and "Model Evaluation" and connects them automatically.

Case Study 1.16

ML in Knime

- Use Predictive Analytics Customers.xlsx and New Customer Prediction.xlsx
- Investigete the data: the marital status, the gender, the estimated yearly income, the number of contracts of this client, the age, then the target is 1 or 0. (target is either a one or a zero and this is actually the column we want to predict later on because it simply points out whether a customer will churn or whether a customer will remain at our company)

Case Study 1.17

- The second file is about a new customer. So this is the one we're going to then try to predict. (only column which is missing in this one here is the target)
- Create a new workflow
- Import the Data
- Need to train our model and then we need to test our model and this is why we normally want to have two datasets, one is for training the model, the second one is for the validation of the model. we need to actually partition our data.
- We need to split the data to training and a validation data set.
- Use the Table Partitioning Node (70-30 or 80-20) (want to have relative and 70% for the training data set and 30% which would be then the second output port, which is the test or validation data set which are often used in the machine learning community.)
- do the stratified sampling based on the target.

Case Study 1.18

Use hte random forest ML

- train machine learning models, you always use the learner nodes. in order to make predictions use the predictor nodes
- drag the random forest learner then connect the data here (Target column – Target (random forest learner can only be a string column) convert if you need, Inculde all the others
- The split criterion By default, the information gain ratio. But you can also try out the Gini index or just information gain. This is also a little bit of trial and error because there are often not one specific way how you have to do it. try and test and figure out what works best for your specific use case.

Case Study 1.18

And for the outputs we have the Out-of-bag bag Out-of-bag Predictions. The first one here, we have attribute statistics as output model. And the most important thing is this well, gray square here because this is the trained model.

(In machine learning, "out-of-bag prediction" (OOB prediction) means evaluating data points that were not included in the training sample of a given model (e.g., decision tree) in bagging.) Out-of-bag (OOB) predictions are a method used in random forests and other ensemble models to estimate prediction error without needing a separate validation dataset. During training, each tree in the forest is built from a random sample of the original data, typically about two-thirds of the total. The remaining one-third, not used in training a given tree, is called the "out-of-bag" data for that tree.

Case Study 1.19

- We want to actually know here now how good is the model, drag here the predictor, Then we connect our train model to the predictor like this and also our other data set, so the test of validation data
- See the predictions percentages
- Then for the total score for the model use the socorer nodes get confusion matrix. It gives us an overview of how good the model was actually containing the predictions.
- 76.9% of the cases, the model predicted the correct result. So that's actually not too good. We need do a little bit more data cleaning and then increase our model, maybe train it a little bit longer, change the settings in the model
- Analyze the new costumer data
- Try another modells (Decision tree, Gradient boosted trees)
- Choose the AutoML node to select the optimal model (AutoML – Worksflow writer) (FROM the Hub) See the results wint F10

Task 1. Titanic - Machine Learning from Disaster

Use machine learning to create a model that predicts which passengers survived the Titanic shipwreck.

(https://www.kaggle.com/competitions/titanic/overview)

Titanic - Machine Learning from Disaster

The sinking of the Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the widely considered "unsinkable" RMS Titanic sank after colliding with an iceberg. Unfortunately, there weren't enough lifeboats for everyone onboard, resulting in the death of 1502 out of 2224 passengers and crew. While there was some element of luck involved in surviving, it seems some groups of people were more likely to survive than others.

Build a predictive model that answers the question: "what sorts of people were more likely to survive?" using passenger data (ie name, age, gender, socio-economic class, etc).

One dataset is titled train.csv and the other is titled test.csv. Train.csv will contain the details of a subset of the passengers on board (891 to be exact) and importantly, will reveal whether they survived or not, also known as the "ground truth". The test.csv dataset contains similar information but does not disclose the "ground truth" for each passenger. It's your job to predict these outcomes. Using the patterns you find in the train.csv data, predict whether the other 418 passengers on board (found in test.csv) survived.

Titanic - Machine Learning from Disaster

- Goal: It is your job to predict if a passenger survived the sinking of the Titanic or not. For each in the test set, you must predict a 0 or 1 value for the variable.

- Metric: Your score is the percentage of passengers you correctly predict. This is known as accuracy.