Introduction to Data Warehouses and Business Intelligence Systems

Jakub Abelski, M.Sc.
J.Abelski@mini.pw.edu.pl

Hurtownie danych i systemy Business Intelligence
Wydział Matematyki i Nauk Informacyjnych
Politechnika Warszawska

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zadanie 10 pn. Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych, realizowane w ramach projektu ,,NERW 2 PW. Nauka - Edukacja - Rozwój - Współpraca'', współfinansowanego ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

Objectives

- Learn why data warehouses are used to:
  - contain key data of enterprises
  - and enable (strategic) decision making in enterprises
- Learn how to plan a data model for data warehouses
- Explore tools and techniques used for developing data warehouses
- Learn why and how data warehouse can be a key data source for advanced analytics
- Learn how to analyse and visualise data using enterprise business intelligence systems
- Gain hands-on-experience with some of key enterprise level platforms

Lecture

1. Introduction to Data Warehouses and Business Intelligence Systems.
2. Data warehouses and dimensional modelling. Normalization and denormalization.
3. Detailed dimensional techniques. The role of time dimension and data changes in the design of the data model.
4. Business process management.
5. Enterprise Data Warehouse Bus Architecture.
6. Data integration - ETL / ELT processes.
7. Business Intelligence systems. Data visualization.
8. DW/OLAP/BI deployment considerations.

Laboratories

1. Introduction to Data Warehouse and Business Intelligence systems.
2. Implementation of sample fact and dimension tables.
3. Advanced data modelling techniques.
4. Preparation of the data cube and visualization of data from the fact and related dimensions tables.
5. ETL process setup and data integration.
6. Metadata management in Business Intelligence systems.
7. Configuration of reports and data visualization in Business Intelligence systems.

Zasady zaliczeń

- Na ocenę składają się punkty z zadań realizowanych w trakcie laboratorium i egzaminu
  - Laboratorium: maks. 60 punktów
  - Egzamin z całości zakresu przedmiotu (w tym całości materiału wykładowego): maks. 40 punktów
- Do zaliczenia wymagane jest co najmniej 50% z każdego z zadań, w tym:
  - Co najmniej 50% punktów z każdej części I zadania laboratoryjnego (tzn. realizowanego w trakcie laboratorium)
  - Co najmniej 50% punktów z II zadania laboratoryjnego (tzn. projektu)
  - Co najmniej 50% punktów z egzaminu
- Ocena końcowa zależy od łącznej liczby punktów:
  - 50-60 => 3.0
  - 61-70 => 3.5
  - 71-80 => 4.0
  - 81-90 => 4.5
  - 91-100 => 5.0

Punktacja części laboratoryjnej

Zadanie	Termin	Punkty
Projektowanie modelu danych hurtowni danych (maks. 15 punktów) Przygotowanie wizualizacji danych w systemie klasy Business Intelligence (maks. 10 punktów)	Laboratorium 9	25
Całościowy projekt obejmujący zaprojektowanie struktur danych, konfigurację procesu ETL i wypełnienie danymi struktur hurtowni danych oraz analiz danych w systemie Business Intelligence (projekt realizowany w okresie tygodni: 8-14)	Laboratorium 15	35

1. Laboratorium obejmuje przykładowe scenariusze realizowane pod nadzorem prowadzącego (całość semestru) oraz konsultacje wspierające realizację projektów (druga część semestru).
2. Dokładne terminy zaliczeń, odpowiedzi i plan zajęć są prezentowane przez prowadzącego zajęcia laboratoryjne.

Kamienie milowe projektu

Kamień milowy	Produkt	Termin złożenia	Punkty
KM0: inicjalizacja projektu	Ustalenie składu zespołów	Laboratorium nr 7	0
KM1: architektura systemu, model przetwarzania i składowania danych	Dokumentacja	Laboratorium nr 11	10
KM2: finalne rozwiązanie	Prezentacja, kod źródłowy, testy raport z projektu	Laboratorium nr 15	25

1. Projekt obejmuje również prezentację wyników poszczególnych etapów prac oraz udzialenie odpowiedzi na pytania od prowadzącego i innych zespołów projektowych.
2. Dokładne wymagania są definiowane przez prowadzących zajęcia laboratoryjne

Zasady formalne przedmiotu

- Udział studentów w zajęciach komputerowych:
  - niepunktowanych nie jest obowiązkowy, aczkolwiek jest zdecydowanie zalecany dla osiągnięcia efektów kształcenia,
  - punktowanych jest obowiązkowy.
- Studenci są zobligowani do usprawiedliwiania nieobecności na zajęciach, na których udział jest obligatoryjny poprzez przedstawienie np. zaświadczenia lekarskiego.
- W końcowej części semestru student może skorzystać z terminu poprawkowego, w trakcie którego może poprawić zadanie punktowane.
- W trakcie pisemnych zadań punktowanych, student może korzystać z własnych notatek, w tym przykładowych własnych rozwiązań analogicznych zadań oraz notatek z wykładu i zajęć komputerowych udostępnionych na stronach wydziału przez osoby prowadzące zajęcia.

Zasady formalne przedmiotu

- W przypadku dostarczenia produktu fazy projektu z opóźnieniem równym 1 tydzień maksymalna liczba punktów za daną fazę zmniejsza się o 20%.
- Opóźnienie większe niż tydzień redukuje maksymalną liczbę punktów do uzyskania za daną fazę projektu o 40%, pod warunkiem finalizacji całości projektu do ostatniego dnia zajęć w semestrze.
- Istnieje możliwość uzyskania dodatkowych punktów za uzyskanie szczególnie ciekawych wyników np. potencjalnie gotowych do zgłoszenia na konferencję.

Wymagania projektowe

- Dwuosobowy, polegający na stworzeniu systemu, w którym:
  - dane z wielu źródeł są integrowane do modelu hurtowni danych
  - przygotowana jest warstwa raportowa danych z wykorzystaniem platformy/platform Business Intelligence
- Idea projektu oraz wykorzystywanych źródeł danych może pochodzić od zespołu i/lub prowadzącego zajęcia
- Projekt powinien wskazywać jasny cel biznesowy (predykcja, badania rynku, analiza problemów) oraz planowane korzyści z punktu widzenia odbiorcy rozwiązania
- Projekt musi posiadać oddzielne warstwy dla przetwarzania ETL (np. SSIS), hurtowni danych (np. MS SQL Server, Oracle) oraz raportowania (np. Oracle Business Intelligence, Power BI, Tableau, SAS Viya)
- Projekt musi obejmować integrację danych pochodzących z co najmniej dwóch źródeł, zbiory danych mogą pochodzić z zasobów internetowych lub zostać wygenerowane sztucznie

Fazy projektu

- Opracowanie koncepcji projektu i wstępnej architektury rozwiązania
- Analiza źródeł danych pod kątem dostępności, harmonogramu odświeżania, niezbędnych transformacji podziału danych
- Projektowanie modelu hurtowni danych uwzględniającego tabele faktowe (dedykowane typy), wymiary i ich zmienność, hierarchie, miary, agregaty, itp.
- Implementacja modelu hurtowni danych
- Przygotowanie mechanizmu pozyskiwania i przetwarzania danych w procesie ETL
- Implementacja warstwy OLAP i BI (źródła, model, hierarchie, kostki, metryki, KPIs)
- Zbudowanie finalnych raportów biznesowych
- Dokumentacja systemu i testy funkcjonalności

Finalna dokumentacja projektu

- Finalna dokumentacja z projektu wymaga następujących elementów:
  - Opis celu projektu oraz planowane korzyści z perspektywy odbiorcy rozwiązania
  - Diagram i opis architektury całego rozwiązania
  - Opis wykorzystanych zbiorów danych
  - Opis transformacji danych w procesach ETL
  - Model hurtowni danych wraz z opisem poszczególnych komponentów
  - Opis warstwy raportowej uwzględniający dostęp do danych, model danych i transformacje w obrębie warstwy raportowej
  - Prezentacja przykładowych raportów dla użytkownika
  - Podsumowanie rezultatów projektu pozwalające ocenić jakość rozwiązania z punktu widzenia biznesu
  - Podsumowanie przeprowadzonych testów funkcjonalnych
  - Opis podziału pracy w zespole (autorstwo poszczególnych rozwiązań: projekt, implementacja, testy, dokumentacja)

Key references for the beginning of the semester

- **[Kimball2016]** Kimball R., Ross, M., The Kimball Group Reader: Relentlessly Practical Tools for Data Warehousing and Business Intelligence, Second Edition, Wiley, 2016
- **[Kimball2013]** Kimball, R., Ross, M., The Data Warehouse Toolkit. The Definitive Guide to Dimensional Modelling, 3rd Ed., Wiley, 2013
- **[Howson2013]** C. Howson, Successful Business Intelligence, Second Edition: Unlock the Value of BI & Big Data, McGrow Hill Education, 2013
- **[Linstedt2015]** D. Linstedt, M. Olschimke, Building a Scalable Data Warehouse with Data Vault 2.0, Morgan Kaufmann, 1st Ed., 2015
  [Silversone2001] Silverstone L., The Data Model Resource Book, Vol. 1: A Library of Universal Data Models for All Enterprises, Revised edition, Wiley, 2001
- **[Adamson2010]** Adamson C., Star Schema The Complete Reference, 1st Ed., McGraw Hill, 2010
- **[Root2012]** Root R., Mason, C., Pro SQL Server 2012 BI Solutions, 1st Ed., Apress, 2012
- **[Alan2014]** Simon A., Enterprise Business Intelligence and Data Warehousing: Program Management Essentials, 1st Ed., Morgan Kaufmann, 2014

Enterprise data
What are the opportunities and challenges?

Data warehouse defined

- Data warehouse (DW) is:
  "*a collection of data extracted from various operational systems, loaded into an operational data store or staging area, then transformed to make the data consistent and optimised for analysis*" [Howson2014]
- Data warehouse is serving the needs of entire organisation. It should provide data needed to easily assess the performance of entire organisation.
- Another aspect is that DW is a database designed for reporting with one or many centralised fact tables, measures and optional supporting dimension tables [Root2012]. This refers more to technical aspects.
- In some cases, DW/BI or DWH/BI abbreviations are used. Still, the data extraction and transformation need to precede the use of visualisation and analytical tools.

Business Intelligence defined

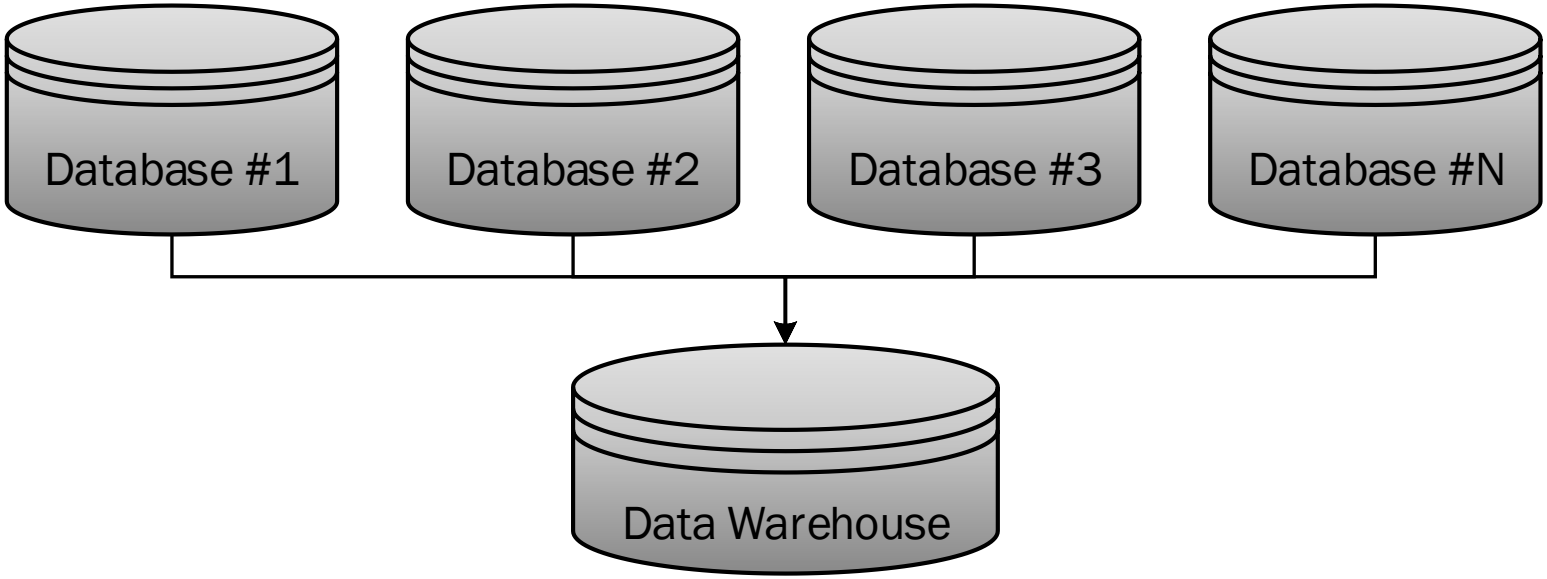
- Business Intelligence (BI) is:
  "*a set of technologies and processes that allow people at all levels of an organisation to access and analyse data*"
  [Howson2014]
- BI tools concentrate on data analysis.
- However, this may be difficult in a large organisation having a lot of databases with not clear integration rules and varied data retention policies.
- Therefore, a data warehouse is a preferred data source for BI tools

Data and information in organisations

- One of the most important assets for any organisation is information. It is used for:
  - Operational purposes i.e. everyday execution of business processes
  - Analytical purposes and decision making
- In line with this division, data is:
  - created in operational systems
  - analysed in DW/BI systems

Studies show that intangible assets (brand, patents, know-how) comprise on most of the value of the largest companies. This is unlike at the beginning of 20th century. DW/BI systems add analytical capabilities to database systems used to serve everyday operational needs such as calculating salaries or managing inventories.

The idea of a data warehouse



Selected data from several databases used by an organisation is extracted, transformed and loaded into a data warehouse. ETL (Extract Transform Load) systems can be applied to handle this process. Typically, there is no need to transfer the data in an online manner as data warehouse is used to observe trends in the data and aggregate values. For this sort of queries having the data from the last hour is not needed.

Data warehouse – typical queries

- Total sales of group of products per a month
- Total income per a company division
- Volume of sales per a country and region
- Rank of products sorted by total income grouped by quarters and years
- The average time of process execution such as Product delivery process in various company regions
- …

Business Intelligence vs. Data Warehouse

- To have a data warehouse does not mean to have a BI.
- A key part of BI deployment are the tools that let users transform data into useful information [Howson2014] [Root2012]. This relies on efficient DW, but also reporting mechanisms
- A data warehouse allows to store and aggregate the data, but may not offer tools for end users
- Hence, **BI is not a synonym of a data warehouse**, even though it relies on data warehouses in most implementations

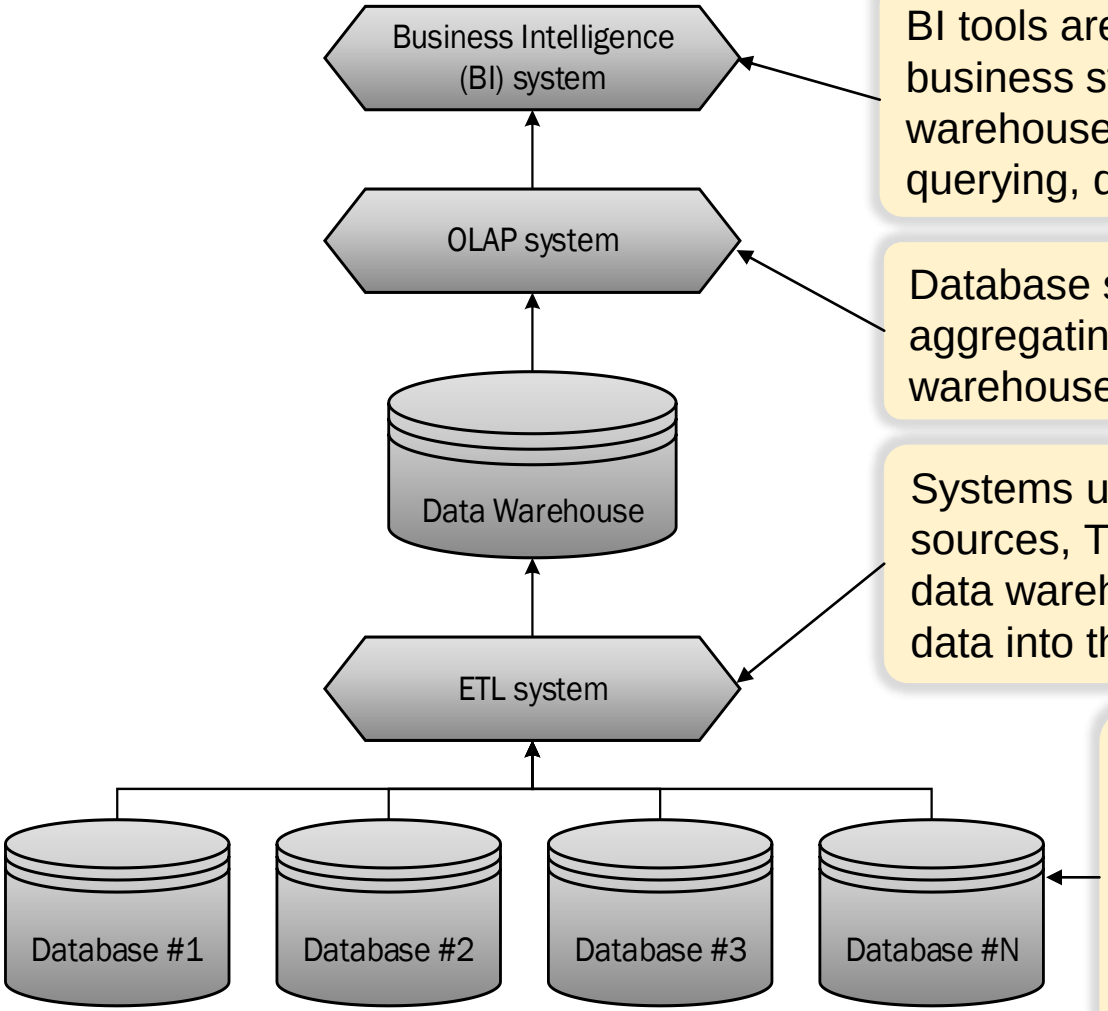
1. Terminology issues affect also data warehouse.
2. Data mart, data silo or data factory are other names used in this context [Root2012].
3. Unlike DW, data mart usually serves the needs of a part of, rather than entire organisation.

OLTP vs. OLAP

- **OLTP – On-Line Transaction Processing**
- OLTP system – a system used to process transactions in an on-line manner i.e. a system serving to manage everyday business data as it appears
- **OLAP – On-Line Analytical Processing**
- OLAP system – a system used to support decisions by analysing and exploring the data stored in a data warehouse. It may use its data structures such as pre-aggregated data to speed up information retrieval.

When OLAP term is used this suggests the system is mostly responsible for data processing, including possible pre-aggregation of data and dedicated forms of aggregated data storage such as data cubes.
BI suggests emphasis on the analysis and exploration, frequently not accompanied by dedicated data storage forms at a storage level.

Complex data flow



BI tools are expected to be used primarily by business staff to navigate through the data warehouse and/or analyse the data (reporting, querying, data analysis and visualization tools)

Database systems that primarily involve aggregating large amounts of data in a data warehouse

Systems used to Extract data from operational sources, Transform data for import into the data warehouse, and Load the transformed data into the data warehouse

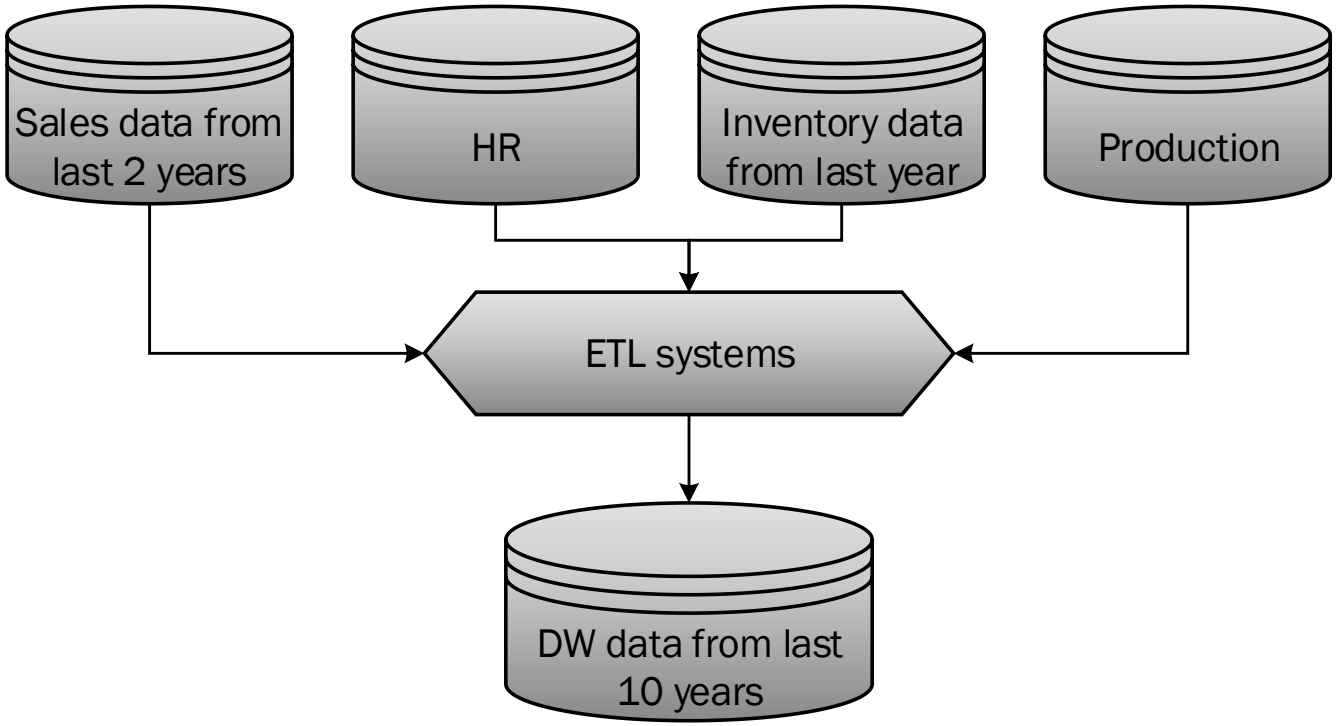
This is where source data is placed in by operational teams such as salespersons, accounting staff and everyone using OLTP system to support his/her everyday operations

Extract-Transform-Load systems

- Extract-Transform-Load (ETL) system is a software system used to transfer the data between different databases and perform all necessary transformations
- In some cases, ELT term is used instead, as some organisations decide to load the data first and transform it next
- Some popular ETL systems include:
  - Informatica PowerCenter
  - Oracle Data Integrator
  - SAS Data Integration Studio
  - Azure Data Factory / AWS Glue / Google Cloud Dataflow
  - Fivetran
- ETL systems handle complex data workflows on regular basis

ETL process can get complex, especially at the transformation stage, which includes aspects such as dealing with missing data, decoding values etc.

Data warehouse and operational databases (example)



There are two major benefits of running a DW:
- Data from longer periods than needed in operational databases can be accumulated
- Data showing the performance of an entire organisation e.g. production combined with inventory can be analysed, unlike in source systems.

OLTP vs. DW/OLAP/BI systems

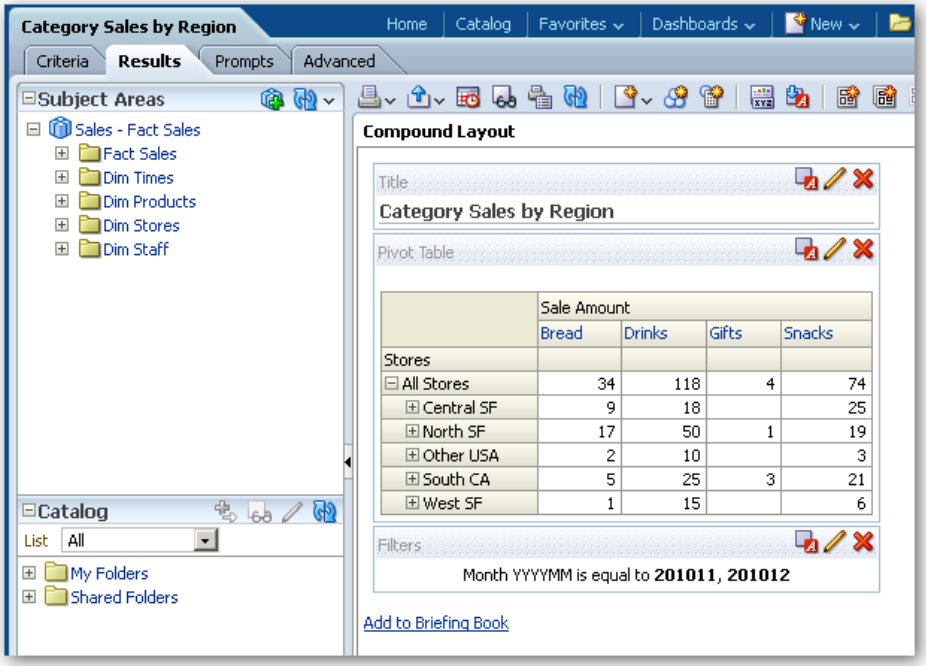
	OLTP	DW/OLAP/BI
Objectives	Efficiently support ongoing operations i.e. execution of business processes	Efficiently support analysis of the data providing basis for strategic decision making
Optimised to	Process transactions quickly	Execute queries quickly
Granularity of the data that particular attention is paid to	Individual records and transactions e.g. individual invoices, orders,…	Aggregated data based on multiple raw facts such as aggregated data on sales and orders
Time aspect	Most recent data and current status of entities e.g. the most recent content of an order is of interest	Data from long periods of times is of interest as it shows trends

Requirements for DW/BI systems according to [Kimball2013]

- **Make information easily accessible**.
  Data structures and labels intuitive for non-developers. High performance of query execution.
- **Present information consistently**.
  Carefully select labels, definitions and named used throughout the system.
- **Adapt to change**.
  Changes are inevitable, but existing applications and data cannot be invalidated by them.
- **Present information in timely way**.
  Find a balance between cleaning new data and providing it within short period of time.

Sample analysis: pivot table

- The report is an example of analysis performed by a BI software. The data may possibly come from various operational systems. A user can easily drill down to see detailed volume of sales of individual products in various countries.
- No knowledge of SQL is needed. Aggregate results may be even pre-computed by the OLAP software to ensure on-the-fly response to user actions.



Big Data defined

- There is no strict definition of Big Data especially in terms of minimal volume of data considered to be big enough to justify the name.
- A frequently accepted definition is that: *"Big data is high-volume, high-velocity and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making"*
  http://www.gartner.com/it-glossary/big-data/
- Innovative forms of information processing (rightly) suggests that Big Data is not handled with BI tools, as these are present on the market for many years.
- Image source:
  https://www.borndigital.com/2015/05/21/big-data-journalism-2015-05-21

Big Data: between true potential and disillusionment

- Big Data is already after disillusionment period.
- Still, there is real phenomenon behind Big Data:
  - More and more aspects of the world are described with data. This includes thoughts expressed at social posts, GPS traces from navigation systems showing communication paths, logs from web servers showing the interests of the visitors, social networks available in digital form, Internet of Things (IoT) streams ...
  - Majority of data is available in digital form, which makes it ready for processing
- Data became truly fundamental for some of the services such as Internet bookshops exploiting the advantages of recommendation systems.

Big Data systems

- There are Big Data software systems that enable efficient storage and processing of large volumes of data (petabytes and more) including:
  - data storage (Hadoop, Hive, HBase, MongoDB, Cassandra),
  - batch processing (Hadoop, Spark),
  - stream processing (Kafka, Spark, Flink, Storm),…
- Finally, Big Data systems are mature enough to have their own best practices defined and architecture guidelines, such as Lambda and Kappa architecture

Big Data platforms will be covered during a separate course.

BI vs. Big Data

Typical settings	Business Intelligence systems	Big Data systems
Volume of data	Gigabytes or terabytes	Terabytes or petabytes
Server layer	Single server or a few servers	Clusters of tens or hundreds of servers
Specialises in	Analysis and visualisation of data	Storage and processing of large volumes of data
Data managed	High quality data extracted from relational databases	Varied content arrived from various sources, most of them being non-relational such as sensor data, unstructured data, web posts, server logs…
Data ingestion	It is acceptable for a BI system to be uploaded with the data from data sources on periodical basis only e.g. once a week	Big Data system is populated with the data constantly, as it is supposed to keep raw data (a.k.a. master data) rather than only data extracted from other sources. Hence, Big Data has to handle larger velocity of data.

Even though Big Data systems are rapidly developed in many organisations, they are not a replacement of BI systems. Each category of the systems has their own role.

Organisation benefits of running BI system

Category	Possible benefits
Management and control	1. Constant access to business metrics/KPIs such as volume of sale, divergence from sales target 2. Ability to easily drill down into details when aggregate metrics do not match expectations
Improved business performance	1. Ability to initiate and observe the results of cross-selling 2. Elimination of unsuccessful marketing campaigns
Benefits from running operational BI	1. Improvements in the works of departments using BI tools through improved decision-making and reduced process cost 2. BI can be used also as an embedded BI i.e. some BI-based reports can be integrated into other applications
Process improvement	1. Analysis of process performance through BI can reveal process bottlenecks e.g. most time-consuming steps or steps requiring rework. As an example, Boeing company uses near real-time dashboards to track assembly of planes
Improved customer service	1. Identify root causes of warranty issues 2. Observe the number of complaints and their reasons

Technical benefits of running DW/BI system

- The ability to integrate and visualise the data coming from different source systems e.g. link the sales data from Oracle database with accounting data from Ms SQL Server database.
- The ability to control the access level to all enterprise data including table/row level security, user groups and scopes.
- The ability to enrich datasets with well-defined metadata and data catalog description, allowing for self-service capabilities
- A more recent capability available in some offerings (SAS Institute, Oracle, …) involves the integration of Big Data systems with BI systems. In such cases, Big Data system may store raw, semi-structured or unstructured content such as social media data. Based on it, useful structured data can be extracted and integrated with other sources of BI systems.

Implementation of DW/BI projects

- Requires:
  - a mixture of various skills and roles covering: Data Architecture, Business Analysis, Data Engineering, Database/BI Development, Data Governance and Catalog, Data Quality, Data Security, etc.
  - support from high level management when organisation-wide view of the data has to be established
- It is the end user, who does not have to be a database expert, that should be provided with appropriate tooling and consistent view of integrated data sources
- Attempt to build in-house BI software is not recommended. The ease of use and access is mandatory for BI software, it means that web-based interface of BI system is preferred.

In practice, the non-technical aspects can be a major challenge. Different departments may have their own databases and data processing methods. They may be reluctant to share their data, integrate it with other sources, adopt standardised solutions such as organisation-wide attribute naming.

Magic Quadrant for Data Integration Tools (22-23)

- Magic Quadrant (MQ) is a series of market research reports prepared by Gartner company. They are built based on proprietary qualitative data analysis methods to visualise market trends including participants, their position, directions and maturity.
- The data integration leaders according to this report are Informatica, Oracle and IBM.

Figure 1: Magic Quadrant for Data Integration Tools

Magic Quadrant for Analytics and Business Intelligence Platforms (22-23)

- There are many MQ reports in the data management area including technology trends, tools and solution providers.
- Microsoft and Tableau are considered the leaders for analytics and BI, while Oracle and SAS are visionaries.

Figure 1: Magic Quadrant for Analytics and Business Intelligence Platforms

Source: Gartner

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zadanie 10 pn. Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych, realizowane w ramach projektu ,,NERW 2 PW. Nauka - Edukacja - Rozwój - Współpraca'', współfinansowanego ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

Data warehouses and dimensional modelling

Jakub Abelski, M.Sc.
J.Abelski@mini.pw.edu.pl

Hurtownie danych i systemy Business Intelligence
Wydział Matematyki i Nauk Informacyjnych
Politechnika Warszawska

Fundusze Europejskie
Wiedza Edukacja Rozwój

Rzeczpospolita Polska

Politechnika Warszawska

Unia Europejska
Europejski Fundusz Społeczny

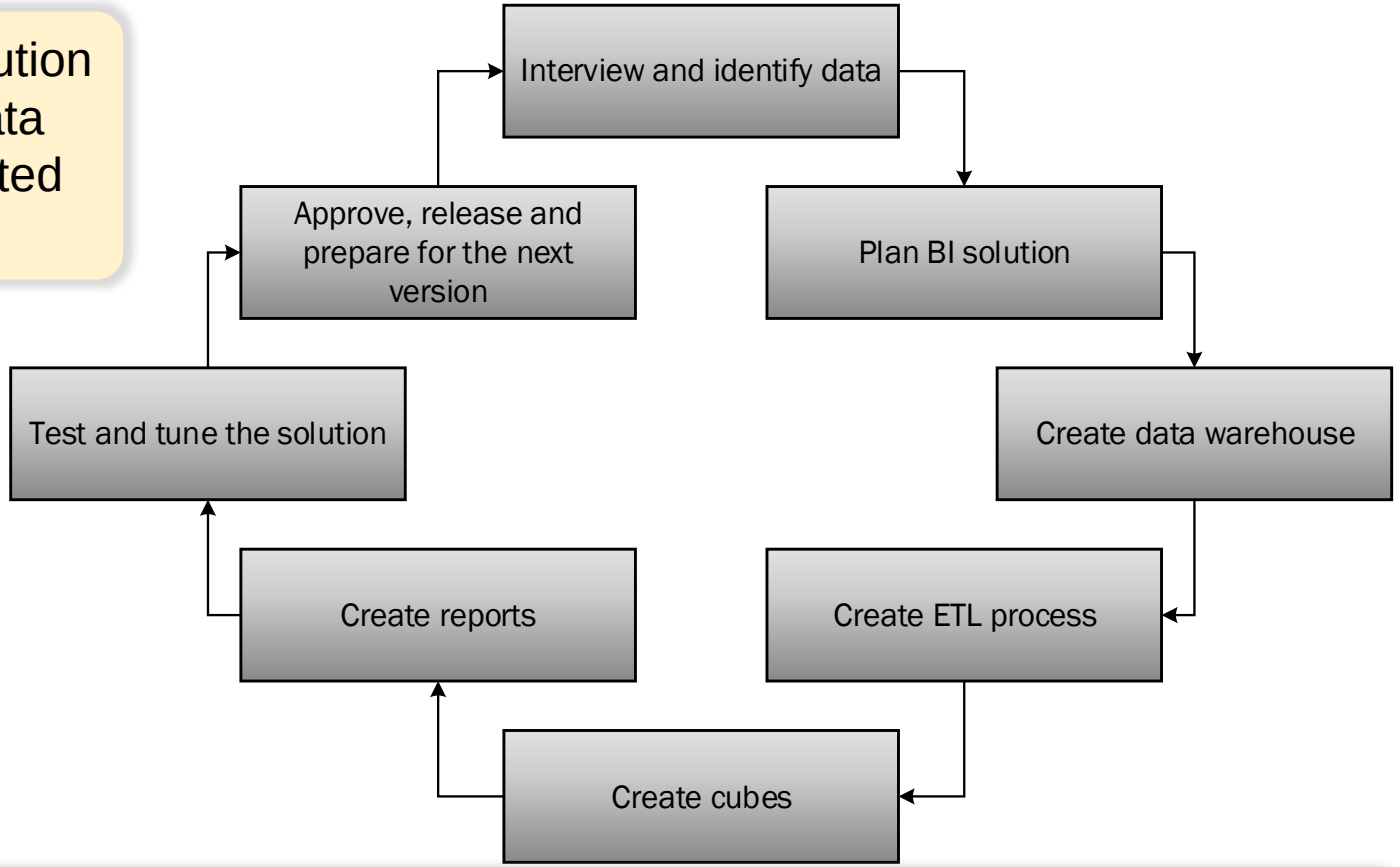
Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zadanie 10 pn. Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych, realizowane w ramach projektu ,,NERW 2 PW. Nauka - Edukacja - Rozwój - Współpraca'', współfinansowanego ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

BI solution life cycle [Root2012]

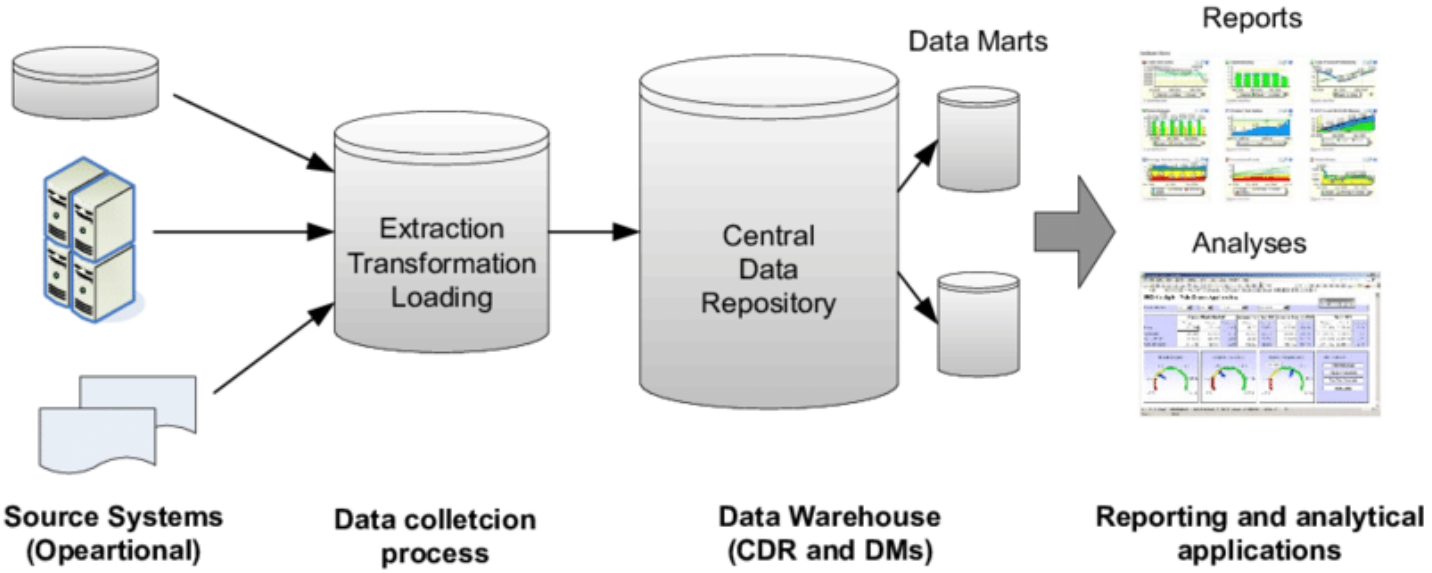
Even before BI solution can be planned, data has to be investigated first.



Equally importantly, creating a data warehouse combined with ETL process is not a final step, since front-end part i.e. interactive data presentation based on Business Intelligence system is needed.

Data warehouse definition revisited

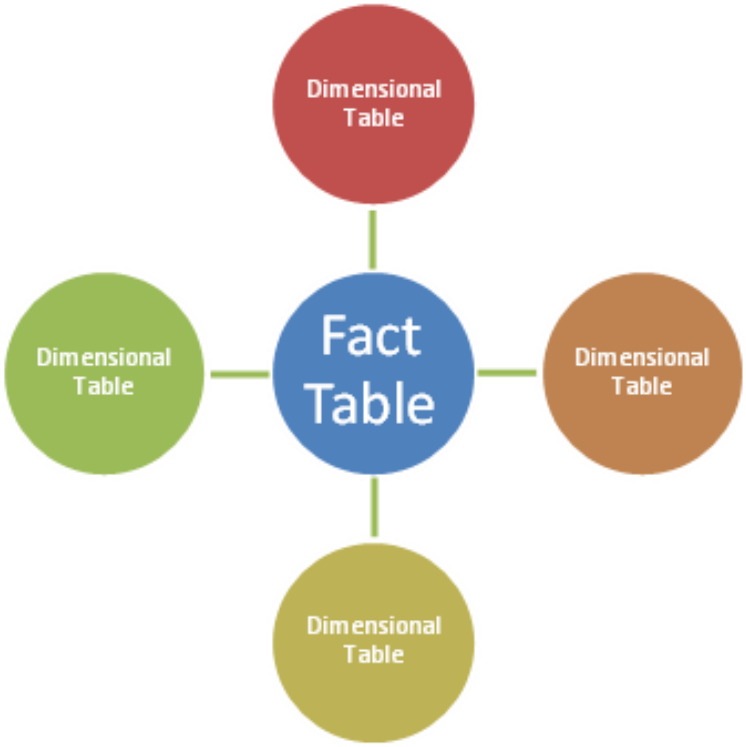
- *"A collection of data extracted from various operational systems, loaded into an operational data store or staging area, then transformed to make the data consistent and optimised for analysis"* [Howson2014]



- Diagram source: https://www.researchgate.net/publication/277476655_Proposal_of_a_New_Data_Warehouse_Architecture_Reference_Model

Data warehouse definition revisited

- Another aspect is that DW is a database designed for reporting with one or many centralised fact tables, measures and optional supporting dimension tables [Root2012]. This refers more to technical aspects.
- In some cases, DW/BI or DWH/BI abbreviations are used. Still, the data extraction and transformation need to precede the use of visualisation and analytical tools.



BI solution levels

- Most frequently BI solution is run at the level of an entire organisation to observe the performance of the organisation across different departments and processes
- However, BI software can be used to run also:
  - operational BI answering detailed everyday questions such as: available inventory, list of malfunctioning devices in a production plant on a particular day
  - process-oriented analysis
- Hence, most frequently data warehouse, but also data mart provides data for BI software

Data warehouse vs. data mart

- Data warehouse covers entire organization with all individual departments
- Data mart contains data for:
  - Single subject area
  - Some business units of an organisation rather than entire organisation
- Hence, independent data mart:
  - Is easier to implement as no cross-organisation project is needed
  - Is with time more expensive to maintain than a DW, especially compared to benefits it generates
  - Generates the risk of creating within an organisation multiple inconsistent data marts using their own concepts, meaning of dimensions, and data processing rules

The sources of data for data warehouses

- Usually, internal operational databases in the company, e.g.:
  - Finance
  - Sales
  - Inventory
  - Supply Chain
  - HR
  - IT
- But also:
  - Data from external vendors e.g. Nielsen, IRI
  - Social media data e.g. Facebook, Twitter/X
  - Clickstream data e.g. product views on the website
  - Data extracted from unstructured content such as text (posts expressing opinions about company products)

Data warehouse – key terms

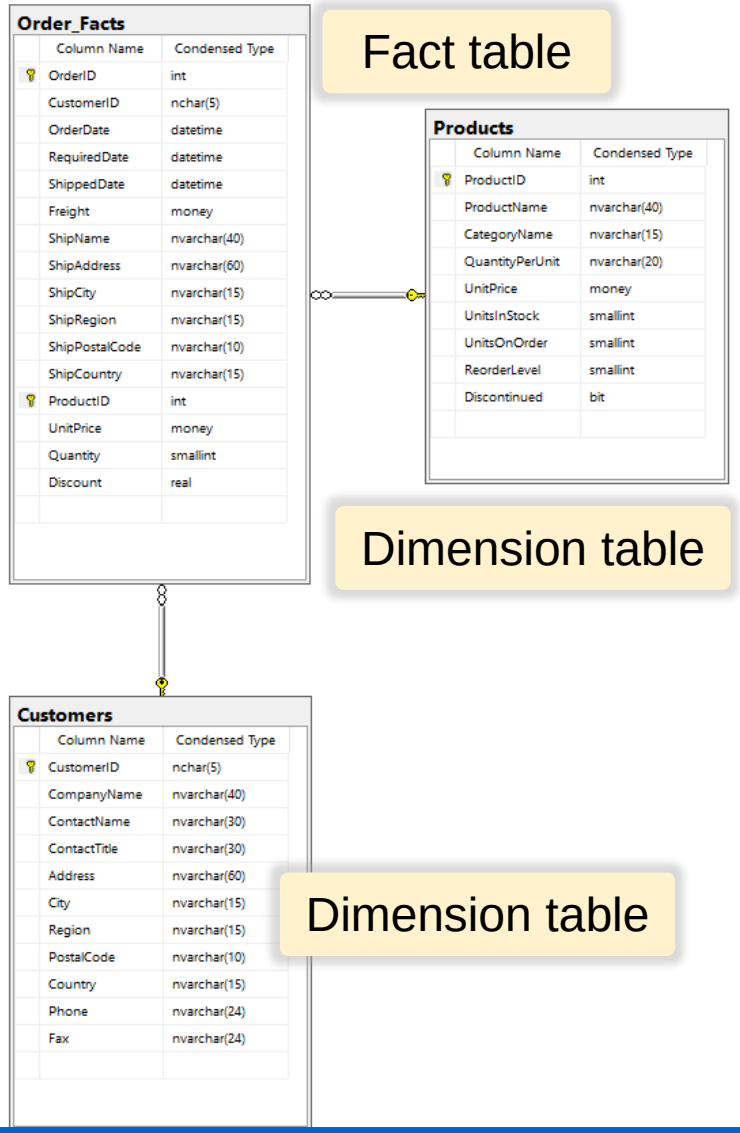
- **Fact** – data element used to measure business performance. Examples:
  - records of sales,
  - purchase,
  - product delivery
- **Dimension** – an attribute or a set of hierarchical attributes, used to group facts. Examples:
  - time (decomposed to year/quarter/month/week/day)
  - location (based on country, region, city and district attributes)
  - organisation (company, branch, division, department)
- **Measure** – typically a numerical attribute of fact, used as a measure of business performance. Examples:
  - Sales in USD
  - Income in USD
  - Number of produced goods
  - Number of labour hours

Data warehouse – key structures

- **Fact table** – a table containing business facts. Typically, the following attributes comprise on the fact table:
  - date of record e.g. date of sales fact, provides for time dimension
  - a number of dimension attributes of foreign keys to dimension tables
  - a number of measure attributes
  - possibly some additional technical metadata
- Fact table usually contains large number of records possibly describing many years of organization's history. Thus, text based descriptive columns, large binary fields and other irrelevant attributes are not present in a fact table.
- **Dimension table** – any related table used to describe the data in facts by providing dimension attributes. Example:
  - Table containing proper city names in case only numerical city identifiers exist in fact table
  - Table containing full product description in case only product codes exist in fact table

Sample simplified dimensional model

- This model is an example of a star schema i.e. a fact table surrounded by dimension tables. Dimension tables in star schema do not refer to other tables.
- Alternatively, a snowflake schema can be implemented. Here, all dimensions are normalized.
- Another option is a galaxy schema also known as fact constellation. In this model multiple fact tables share the same dimension tables. The arrangement of fact and dimension tables forms a collection of stars.



Fact table details - part I

- Store atomic fact data.
- All facts must be at the same level of detail e.g. selling a certain product to a certain client based on a single order.
- Putting into the same table both individual orders and the overall volume of sales to the client within a month is not correct. Aggregate operations will be negatively affected then.
- Fact table usually consume even 90% of a total space allocated by a DW.
- Considering the size of a fact table, it is usually partitioned by a well-defined partitioning key.

Fact table details - part II

- Have a composite primary key i.e. a primary key composed of more than one column (on the logical level).
- Can be denormalised if in the source system the fact related data is maintained in multiple tables.
- Usually, include:
  - At least one date column, which is a foreign key to date dimension table
  - Dimension attributes i.e. references to dimension tables
  - Measure attributes i.e. columns providing basis for measure calculation. These can be:
    - Additive such as order value
    - Semi-additive such as inventory value in daily snapshot of an inventory
    - Non-additive such as unit price
- For some dimensions, there is no associated dimension table. All the data is contained in an attribute such as OrderId. These dimensions are called degenerate dimensions or fact dimensions.

Example simplified fact table

Degenerate dimension aka. fact dimension

Dimensional attribute (reference to dimension table)

Fact dimensions, but ideally should be converted to a dimensional attributes

Measure (numerical i.e. money data type)

Fact dimensions, but ideally should be extracted into a sperate dimension

Dimensional attribute (reference to dimension table)

Measures (data grain should be considered)

Order_Facts

	Column Name	Condensed Type
🔑	OrderID	int
	CustomerID	nchar(5)
	OrderDate	datetime
	RequiredDate	datetime
	ShippedDate	datetime
	Freight	money
	ShipName	nvarchar(40)
	ShipAddress	nvarchar(60)
	ShipCity	nvarchar(15)
	ShipRegion	nvarchar(15)
	ShipPostalCode	nvarchar(10)
	ShipCountry	nvarchar(15)
🔑	ProductID	int
	UnitPrice	money
	Quantity	smallint
	Discount	real

The not obvious treating of date attributes

- Defining date attributes as a fact dimension is not the best practice. Instead, a date dimension table is usually defined. This may seem to include obvious attributes only (year, month, day). However, the Date dimension contains also:
  - Public holiday markers
  - Fiscal year assignment
  - The names of public holidays (Easter, Independence Day, ….)
- The primary key for Date dimension table can be:
  - Sequentially assigned Integer
  - Meaningful key, such as an integer representing YYYYMMDD
  - Date or DateTime
- Why not put all these attributes in fact tables?
  - Well, there are many dates and (surprisingly) many attributes describing date dimension

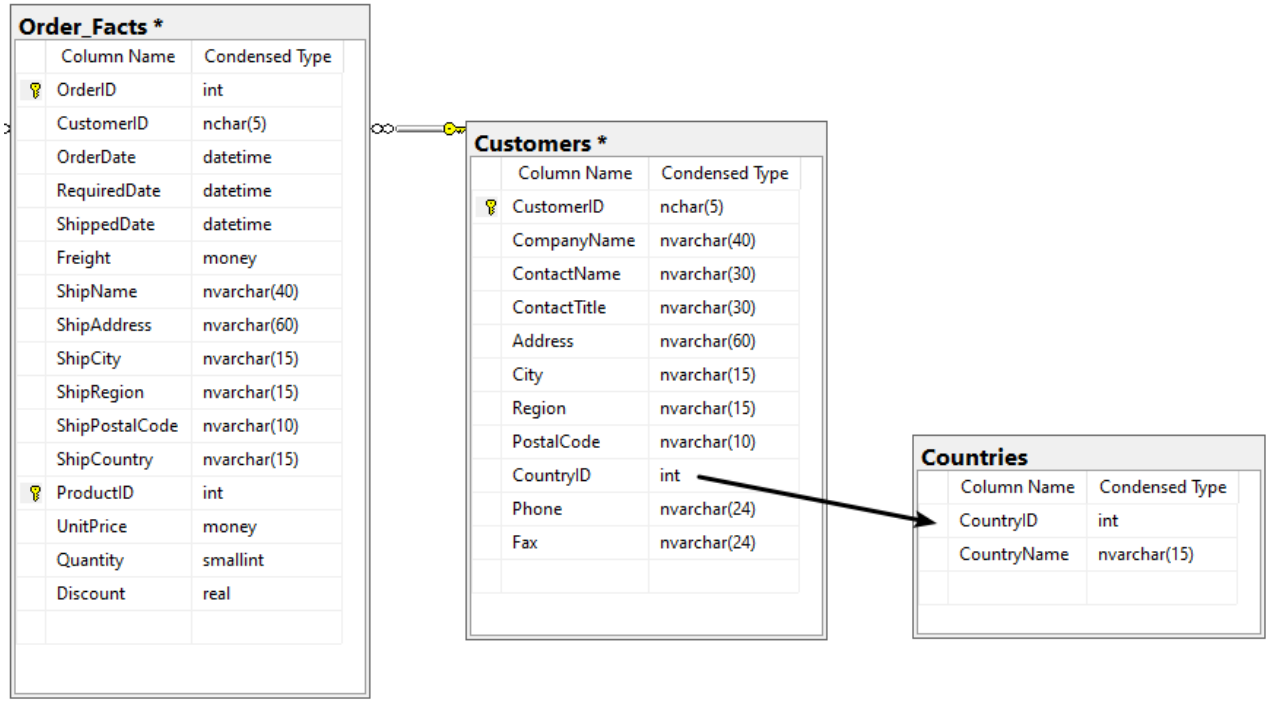
Date dimension table is an example of role-playing dimension. The table can be linked to variety of dimension keys such as Order Date, Delivery Date, …

Dimension table details

- Has a primary key composed of a single column
- In real deployments can have even 50-100 columns
- Are recommended to be denormalised i.e. in general do not refer to other tables
- Provide labels and selectors for reporting needs
- Example:
  - Order_facts table contains just CustomerId i.e. identifier most likely not clear for an end user of a BI system
  - Customers table contains company name and detailed contact info

Dimensions - star schema vs. snowflake schema

- When dimension tables are normalised, we get a snowflake schema. Such a schema typically more closely resembles underlying model of a relational database.
- However, the main drawback of snowflaking is that we get too complex data model, especially for a business end users. The simplicity of a a star schema having denormalised dimension tables is a major advantage.



Some dictionary tables could be developed for city, region, country, postal code and referenced via foreign keys from customer's table.

A complex database with relations linking city and country tables could be even developed.

The role of attribute naming

- Attribute names must be clear not only for IT staff, but even more importantly for business users.
- All codes such as 01 denoting "orders delivered" should be decoded to text labels.
- All composite codes should be decoded. Example:
  - codes such as EMEA/SW should be decoded to atomic attributes
  - for instance, EMEA/SW can be a composition of:
    - Region (with one of the values being EMEA - standing for Europe, Middle East and Africa)
    - Market (with one of the values being SW - standing for Software)
- Appropriate naming and decoding can be applied at DW and/or BI level.
- The naming convention should be standardized to define recommended naming patterns, e.g. the use of Pascal case, standardized prefixes and suffixes, dictionary of abbreviations.

Joins in a data warehouse

- Typically, fact tables are joined with dimension tables only.
- However, there can be multiple fact tables e.g. orders and sales plans. The relation between them may be not straightforward:
  - Some of the plans may be not reflected in orders.
  - Some orders may be not expected i.e. planned.
- In general, fact to fact joins should be avoided.
- Instead, fact-dimension-fact joins are suggested:
  - Example: orders – customers – sales_plans.
  - Full outer join is used to combine participating tables.
  - This technique is named drill-across, multipass, multi-select, multi-fact or stitch queries.

Oracle Business Intelligence uses the term „stitch join”. Different names are used by different DW/BI tools.

Different categories of fact tables

Category	Definition	Features
Transaction	Each fact is a single real-life atomic event e.g. delivering a product to client premises on a certain day	• Most frequently used. • Fact records are added, but usually not updated later.
Periodic snapshot	Each fact is a snapshot of the data at a certain detail performed on regular basis e.g. daily snapshot of the inventory showing the amount of each product in the inventory at the end of each day	• Facts are semi-additive i.e. cannot be added over time as this would be meaningless • However, such facts can be averaged e.g. to show the average number of products in stock. • Can generate large tables since snapshots are supposed to be created also for out-of-stock products. • Fact records are added, but not updated later.
Accumulating snapshot	In this case, one fact can be created for every proces instance e.g. for one product order. Next the fact is updated as the order status changes from submitted to delivered.	• The performance of individual process instances can be efficiently tracked e.g. to calculate the average number of hours between submitting an order and deciding about delivery date. • In this case, fact record is updated as the process proceeds. This can be difficult for OLAP cubes.

Sample periodic snapshot fact table (Fact_Order)

ProductKey	CustomerKey	Date	Quantity	…
101	ABC	2023/12/22	10	…
103	ABC	2023/12/22	4	…
103	DEF	2023/12/22	4	…
…	…	…	…	…
103	ABC	2023/12/23	12	…
104	DEF	2023/12/23	6	…
…	…	…	…	…

When a new order is created in the source system it is later added to the fact table during the ETL process (e.g. once a day). When some orders are updated or cancelled, then the fact table in DW will need to reflect those changes. New data could be merged into the fact table, or each version of data could be kept as a separate row. Depending on the decision, selection of distinct orders will need to be implemented in a different way.

Sample periodic snapshot fact table (Fact_Inventory_Daiy)

ProductID	Date	Quantity	…
101	2023/12/22	25	…
103	2023/12/22	4	…
…	…	…	…
101	2023/12/23	0	…
103	2023/12/23	12	…
…	…	…	…

For every day, for every product the quantity of the product at the end of the day in company inventory is saved.

We can easily find out that for some products, on same days, we run out of supplies.

The data contained in snapshot table is redundant i.e. we could easily get the same numbers from transactions such as Goods Issued and Goods Received documents. However, there can be hundreds of such documents per a day. Hence, it may be inefficient to analyse such raw data in some cases.

Sample accumulating snapshot fact table (Fact_Delivery)

OrderID	Submit_date	Confirmed_date	Lag_submit_to_confirmed	Ready_for_delivery_date	Lag_submit_to_ready	…
101	2023/12/22	2023/12/24	2	2024/01/01	10	…
103	2023/12/22	2023/12/22	0			…
…	…	…				…

For every process execution (here: delivery of every order) we track the performance of individual stages

We can easily find out that for some orders, it took us too much time to perform some steps

A row in an accumulating snapshot fact table summarizes the measurement events occurring at predictable steps between the beginning and the end of a process. When analyzing a line on an order, it is initially inserted when the order line is created. As pipeline progress occurs, the accumulating fact table row is revisited and updated. Often the fact table definition is extended with numeric lag measurements and milestone completion counters.

Dimensional model

- The organisation of a DW involving fact and dimension tables is the core of a dimensional model
- The dimensional model:
  - Makes it possible to store and analyse the same data as in a normalised model of OLTP systems
  - Improves understandability of data compared to complex graphs of normalised tables
  - Improves query performance by eliminating the need for majority of join operations
  - Isolates (to large extent) dimensional model used by DW/BI users from changes in the data models of operational systems

Conceptual / logical / physical data model consideration

- Data modeling is the process of defining a data model for the data to be stored in a database. It ensures that required data objects are accurately represented and connected.
- Usually, there are 3 different types of data models:
  - **Conceptual**: defines key elements of the system from the business perspective. Organizes and defines main concepts and rules (typically created by Business stakeholders and Data Architects)
  - **Logical**: defines how the system should be implemented regardless of the underlying database environment (typically created by Business Analysts and Data Architects)
  - **Physical**: defines how the system will be implemented in a specific DBMS (typically created by DBA and developers or generated automatically from the logical data model).
- Data models can be designed with the help of dedicated data modelling tools (e.g. Erwin Data Modeler, DbSchema, ER/Studio). They usually offer a rich set of features including standard notations, backward/forward engineering.

DW software offerings

- There are two options to consider when creating a data warehouse:
  - Use RDBMS same as for operational databases also for creating a DW. For instance, Oracle Database can be used in this role (possibly including Oracle Exadata Machines)
  - Use DW-oriented platform such as:
    - Amazon Redshift
    - Oracle ADW
    - Google BigQuery
    - Snowflake
    - IBM DB2
    - SAP BW/4HANA

One more option is buying DW appliance combining hardware and software such as Oracle Exadata Database Machine servers with Oracle Database software installed. In this way, the complexity of software installation and hardware/software tuning is eliminated. However, recently this approach is being replaced by cloud subscriptions.

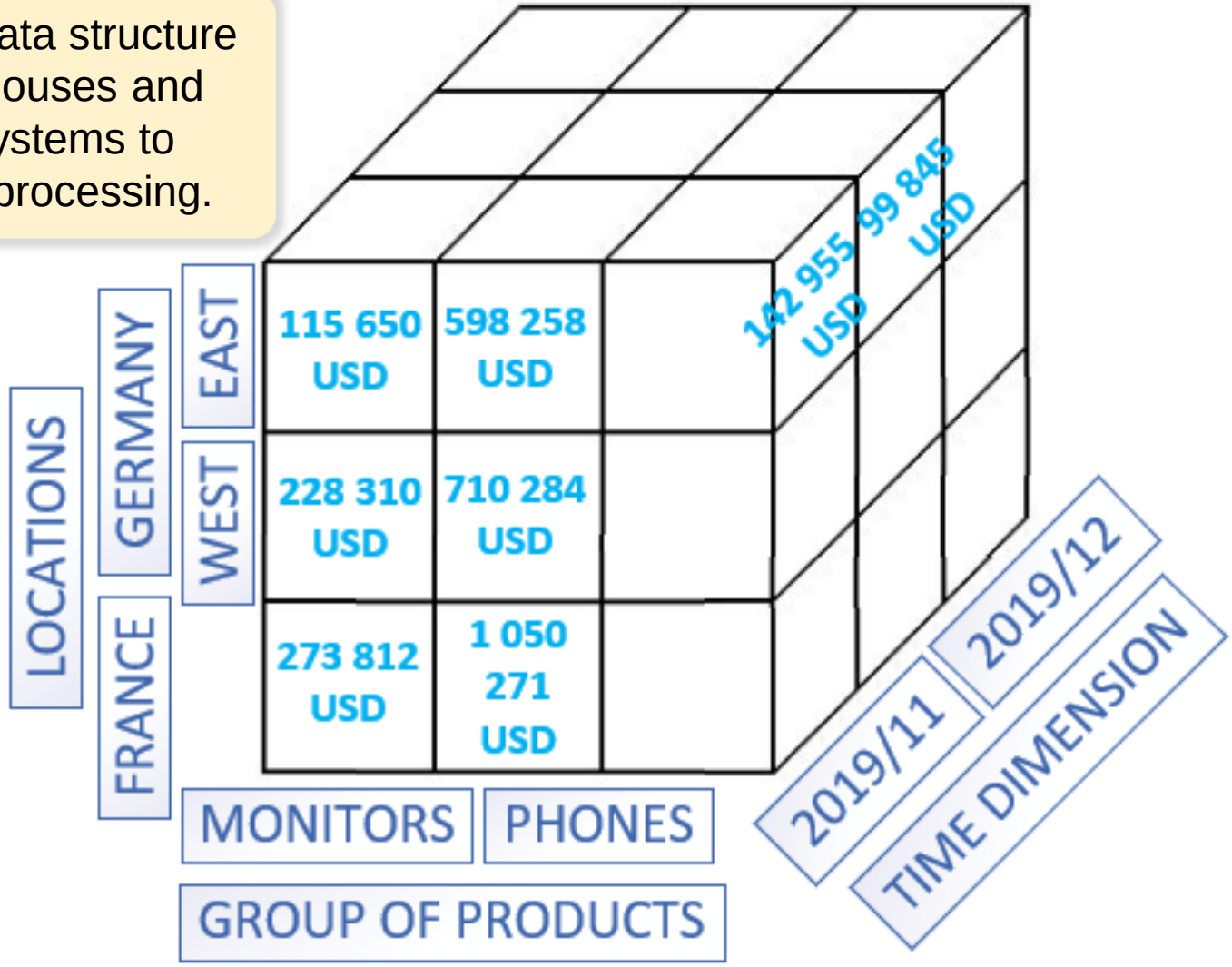
The implementation of dimensional data model

- Dimensional model can be implemented in:
  - Relational DBMS using star/snowflake schemas
  - Multidimensional database environments using OLAP data cubes
- Typically:
  - First, the data is placed in a star/snowflake schema
  - Next, it is used to populate OLAP data cubes

Some BI tools may refrain from creating OLAP cubes but provide similar visualisation capabilities in the form of pivot tables. Hence, OLAP cubes are not mandatory in all BI deployments. However, dimensional model incl. facts and dimensions is still used.

OLAP data cube example

A cube is a primary data structure defined in data warehouses and managed by OLAP systems to ensure efficient data processing.



Data cube – definition

- A multi-dimensional representation of business data:
  - Each cell represents measure value
  - Each edge denotes dimension attribute
  - Cube is built on the table of facts and some dimension tables
- In reality:
  - Frequently more than 3 dimensions
  - More than one measure
  - A number of hierarchical dimensions allows to drill down any part of a cube to find the data of interest e.g. total sales of LCD monitors made in November 2019 in Berlin
  - Can be built on the top of fact tables of all types and their dimension tables

Measures

- For a combination of measure and dimension(s) a specific aggregation method can be defined. The most frequently used one is a **sum**. Remaining methods include:
  - Average
  - Weighted average
  - Weighted sum
  - Minimum
  - Maximum
  - First
  - Last
  - Count

Notice: every time the aggregate function will be calculated for all the facts in a cell, for instance: the number of sales records for Germany in December 2019. A fact could be invoice detail record that refers to one product or order detail record in case orders are considered.

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zadanie 10 pn. Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych, realizowane w ramach projektu ,,NERW 2 PW. Nauka - Edukacja - Rozwój - Współpraca'', współfinansowanego ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

Detailed dimensional techniques

Jakub Abelski, M.Sc.
J.Abelski@mini.pw.edu.pl

Hurtownie danych i systemy Business Intelligence
Wydział Matematyki i Nauk Informacyjnych
Politechnika Warszawska

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zadanie 10 pn. Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych, realizowane w ramach projektu ,,NERW 2 PW. Nauka - Edukacja - Rozwój - Współpraca'', współfinansowanego ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

Introduction

- Developing a dimensional data model means to resolve a number of issues
- The core categories of problems to solve are:
  - *How to track changes in the data?*
    Most OLTP system concentrate on the most recent content of the records e.g. most recent customer data. In the case of DWH solutions, change tracking is a major objective.
  - *How to make data model clear for non-IT users?*
    OLTP data model is usually supposed to be used only/mostly by IT team. This is unlike BI solution.
  - *How to ensure high performance of analytical tasks?*
- This lecture aims on highlighting key answers to these questions
- The content of this lecture is largely based on:

  [Kimball2013] Kimball, R., Ross, M., The Data Warehouse Toolkit. The Definitive Guide to Dimensional Modelling, Wiley, 3rd Ed., 2013

Date dimension continued

- Requires its own dimension table
- The date dimension table is referenced from other fact and possibly dimension tables via foreign keys
- The content of date row, once created does not change
- Hence, rows are only added to the table

SQL | Result

```
SELECT * FROM "_SYS_BI"."M_TIME_DIMENSION" LIMIT 50
```

	DATETIMESTAMP	DATE_SQL	DATETIME_SAP	DATE_SAP	YEAR	QUARTER	MONTH	WEEK	WEEK_YEAR	DAY_OF_WEEK	DAY	HOUR	MINU
1	Jan 1, 0001 12:00:00 AM.000	Jan 1, 0001	00010101000000	00010101	0001	01	01	01	0001	05	01	00	00
2	Dec 31, 9999 12:00:00 AM.000	Dec 31, 9999	99991231000000	99991231	9999	04	12	52	9999	04	31	00	00
3	Jan 1, 1900 12:00:00 AM.000	Jan 1, 1900	19000101000000	19000101	1900	01	01	01	1900	00	01	00	00
4	Jan 1, 2007 12:00:00 AM.000	Jan 1, 2007	20070101000000	20070101	2007	01	01	01	2007	00	01	00	00
5	Jan 2, 2007 12:00:00 AM.000	Jan 2, 2007	20070102000000	20070102	2007	01	01	01	2007	01	02	00	00
6	Jan 3, 2007 12:00:00 AM.000	Jan 3, 2007	20070103000000	20070103	2007	01	01	01	2007	02	03	00	00
7	Jan 4, 2007 12:00:00 AM.000	Jan 4, 2007	20070104000000	20070104	2007	01	01	01	2007	03	04	00	00
8	Jan 5, 2007 12:00:00 AM.000	Jan 5, 2007	20070105000000	20070105	2007	01	01	01	2007	04	05	00	00
9	Jan 6, 2007 12:00:00 AM.000	Jan 6, 2007	20070106000000	20070106	2007	01	01	01	2007	05	06	00	00
10	Jan 7, 2007 12:00:00 AM.000	Jan 7, 2007	20070107000000	20070107	2007	01	01	01	2007	06	07	00	00
11	Jan 8, 2007 12:00:00 AM.000	Jan 8, 2007	20070108000000	20070108	2007	01	01	02	2007	00	08	00	00
12	Jan 9, 2007 12:00:00 AM.000	Jan 9, 2007	20070109000000	20070109	2007	01	01	02	2007	01	09	00	00
13	Jan 10, 2007 12:00:00 AM.000	Jan 10, 2007	20070110000000	20070110	2007	01	01	02	2007	02	10	00	00
14	Jan 11, 2007 12:00:00 AM.000	Jan 11, 2007	20070111000000	20070111	2007	01	01	02	2007	03	11	00	00
15	Jan 12, 2007 12:00:00 AM.000	Jan 12, 2007	20070112000000	20070112	2007	01	01	02	2007	04	12	00	00

```
Statement 'SELECT * FROM "_SYS_BI"."M_TIME_DIMENSION" LIMIT 50'
successfully executed in 2 ms 187 µs  (server processing time: 0 ms 81 µs)
Fetched 50 row(s) in 2 ms 97 µs (server processing time: 0 ms 3 µs)
```

Date dimension – possible columns to be used in the dimension table

Column	Data type	Comments [possible entries]
DATEID	CHAR	Consistent date format, e.g. YYYYMMDD [e.g. 20191224]
DATE	DATE	Same as DATEID but in DATE format [e.g. 24/12/2019]
MONTH_NAME	CHAR	[January, …, December]
DAY_OF_WEEK_NAME	CHAR	[Monday, …, Sunday]
QUARTER_NUM	INT	[1, …, 4]
MONTH_NUM	INT	[1, …, 12]
WEEK_NUM	INT	[1, …, 53]
DAY_OF_WEEK_NUM	INT	[1, …, 7]
DAY_OF_MONTH_NUM	INT	[1, …, 31]
IS_WEEKDAY	CHAR	True for Monday-Friday, false otherwise [e.g. "X" or empty (SAP HANA)]
IS_HOLIDAY	CHAR	Holiday or working day [e.g. "X" or empty (SAP HANA)]
HOLIDAY_NAME	CHAR	[e.g. Independence Day]

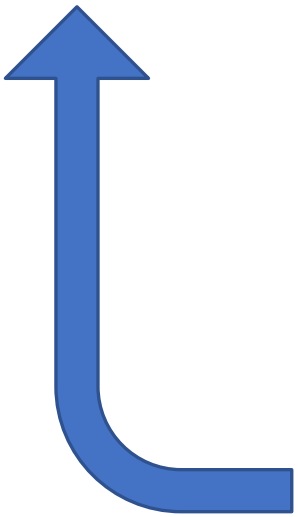
There are other possible entries as well, such as day of quarter, month of quarter, first and last day of month, fiscal year and month etc. Moreover, unknown entry can be also placed in the table.

Role-playing dimensions

- Single dimension table can be referenced many times from other tables to provide data for columns playing different roles. Such a dimension is named role-playing dimension. Examples:
  - Date dimension table can be used in the context of: date of birth, delivery date, payment due date and many other roles
  - Customer dimension table can be used in the context of: client, vendor, manufacturer, servicing company,…
- In such cases, a recommended solution is to:
  - Have a single dimension table
  - Create multiple SQL views based on the content of the same table such as:
    - Delivery Date Dimension
    - Payment Due Dimension
  - Define different column names in every view to avoid ambiguity when column names are dragged and dropped in BI tools
  - Link the views to individual fact tables containing e.g. Delivery Date

Role-playing dimensions illustrated

Delivery_date	Delivery_month	Delivery_year	Delivery_quarter	…
20191210	December	2019	4	
…	…	…	…	



DIM_Delivery_Date view
defined through
CREATE VIEW
DIM_Delivery_Date as
SELECT
    Date as 'Delivery date',
…
FROM DIM_Date

Date	Month	Year	Quarter	…
20191210	December	2019	4	…
…	…	…		

DIM_Date table
Role playing dimension source

Outrigger dimension

- **Outrigger dimension** refers to the situation when an attribute of a dimension table refers to another dimension table.
- Example:
  - Let us consider customer dimension table
  - This may include date of birth column
  - Date of birth can be a reference to Date dimension table

Outrigger *- a piece of wood shaped like a small, narrow boat, that is fixed to the side of a boat (…) to prevent it from turning over in the water.*
Longman Dictionary of Contemporary English, 1990

Outrigger dimensions illustrated

DIM_Customer table

Id	First_name	Last_name	Date_of_birth	…
123	W*****	M*****	19901101	…
…	…	…	…	…

Outrigger dimension are allowed but should not be used too frequently as this reduces the readability of the model. They can be frequently replaced with an additional foreign key in a fact table. For instance, we could potentially imagine placing a foreign key representing client's date of birth in Facts_orders table.

Foreign key

Primary key

DIM_Date table

Date	Month	Year	Quarter	…
19901101	November	1990	4	…
…	…	…	…	…

Degenerate dimension - revisited

- **Degenerate dimension** is the dimension not having its own dimension table. It is a fact attribute that do not fit into any single dimension table, and is stored in the fact table directly.
- Degenerate dimension can be used for data filtering and grouping when processing the fact data.
- Example:
  - Order_id i.e. order identifier present in Fact_orders is a useful dimension attribute
  - However, there is no other data linked to it to be placed out of Fact_orders
  - Hence, Order_id is a degenerate dimension

Interestingly, degenerate dimension, despite its name is not an error. Moreover, it is not something to avoid, but to identify during the data modeling phase.

Junk dimension

- Transactional processes typically produce many different low-cardinality flags and indicators.
- Instead of creating separate dimensions for each attribute, a single junk dimension can be defined.
- This dimension could contain a Cartesian product of all possible values, but it should rather contain the combination of values that actually occur in the source data.

DIM_Customer_Profile (junk dimension)

CustomerProfileID	ProspectFlag	LeadFlag	CustomInd	ExternalInd	…
123	YES	NO	NO	YES	…
…	…	…	…	…	…

Null values in dimension tables

- Null values should not be shown to end users, as this may be unclear
- Instead, values such as ‘Unknown' or 'Not applicable' should be used
- What should be noted here is that in fact there can be many reasons of not having a proper data. Example:
  - Instead of having NULL in graduate date for students who have not graduated yet, we can use a reference to Date dimension record named 'Not graduated yet'
- Interestingly, this technique can be applied not only to character data types, but also to other data types, e.g. some pre-defined dates, numbers, short codes

Surrogate dimension keys

- A **surrogate dimension key** (SKID) is a primary key in a dimension table with artificially generated values
- Key idea: it may be better/necessary to use surrogate primary keys in dimension tables rather than underlying natural primary key values present in operational databases
- Key reasons for surrogate PK in dimension tables:
  - The underlying natural keys may come from different databases and be incompatible. For example:
    DIM_Customers table can be populated with client and vendor records coming from underlying databases, which have partly overlapped values of their PKs
  - For one natural key there can be multiple dimension rows when changes are tracked over time. For example:
    we may want to keep many records for one customer showing the data of this customer for different periods of time

Surrogate dimension keys

- Other reasons for surrogate PK in dimension tables are as follows:
  - Surrogate keys are durable i.e. do not change with time. Natural keys may change with time and/or be re-used. For instance, person identifier used by HR department may be reused after the person is retired
  - There can be many records in dimension table for one record in underlying operational table (see changing dimension table techniques for details)

A surrogate dimension key often takes the form of an integer column containing a sequence of numbers, having no relation to natural primary keys used in a real world. However, this approach requires the dimension data to be loaded before facts to keep a consistency of foreign keys. Interesting option is the usage of **hash keys**, which are generated based on a predefined set of columns ensuring the data uniqueness. This allows the fact data to be loaded independently from dimensions, but still the consistency of the final set of rows should be verified.

Changing dimensions

- In the case of many dimensions, the content of a dimension record may change over time. Examples:
  - Customer dimension
    - Customer address may change with time
    - Customer age will change with time
  - Product dimension
    - Product originally belonging to 'Innovative Products' category can be reassigned to 'Popular choices' category

Dealing with changing dimensions
SCD Type 0: keep original data

Customer table in OLTP

Id	First_name	Last_name	City	…
123	W*****	M*****	Warsaw	…
…	…	…	…	…

DIM_Customer table in DW

Id	First_name	Last_name	City	…
123	W*****	M*****	Warsaw	…
…	…	…	…	…

Changes on 2021/12/22

Customer table in OLTP

Id	First_name	Last_name	City	…
123	W*****	M*****	Krakow	…
…	…	…	…	…

DIM_Customer table in DW

Id	First_name	Last_name	City	…
123	W*****	M*****	Warsaw	…
…	…	…	…	…

When the data of a dimension changes in operational database, it remains as it was originally placed in a dimension table. We just ignore changes in operational DB.

Dealing with changing dimensions
SCD Type 1: overwrite

Customer table in OLTP

Id	First_name	Last_name	City	…
123	W*****	M*****	Warsaw	…
…	…	…	…	…

DIM_Customer table in DW

Id	First_name	Last_name	City	…
123	W*****	M*****	Warsaw	…
…	…	…	…	…

Changes on 2021/12/22

Changes on 2021/12/22+

Customer table in OLTP

Id	First_name	Last_name	City	…
123	W*****	M*****	Krakow	…
…	…	…	…	…

DIM_Customer table in DW

Id	First_name	Last_name	City	…
123	W*****	M*****	Krakow	…
…	…	…	…	…

When the data of a dimension changes in operational database, in a dimension table, the old data is also overwritten in DWH (previously existing data is lost).

Dealing with changing dimensions
SCD Type 2: track changes

Customer table in OLTP

Id	First_name	Last_name	City	…
123	W*****	M*****	Warsaw	…
…	…	…	…	…

DIM_Customer table in DW

SkId	Id	First_name	Last_name	City	…
1032	123	W*****	M*****	Warsaw	…
1037	…	…	…	…	…

Changes on 2021/12/22

Changes on 2021/12/22+

Customer table in OLTP

Id	First_name	Last_name	City	…
123	W*****	M*****	Krakow	…
…	…	…	…	…

DIM_Customer table in DW

SkId	Id	First_name	Last_name	City	…
1032	123	W*****	M*****	Warsaw	…
1037	123	W*****	M*****	Krakow	…

When the data of a dimension changes in operational database, in a dimension table, another record is added. A surrogate key needs to be used here.

Dealing with changing dimensions
SCD Type 2: track changes - details

DIM_Customer table in DW

SKID	OrgID	First_name	Last_name	City	Valid_from	Valid_to	Status
1033	123	W*****	M*****	Warsaw	19700101	99991231	Curr
…		…	…	…			

Changes on 2021/12/22+

DIM_Customer table in DW

SKID	OrgID	First_name	Last_name	City	Valid_from	Valid_to	Status
1033	123	W*****	M*****	Warsaw	19700101	20211221	Hist
1037	123	W*****	M*****	Krakow	20211222	99991231	Curr

In this case, a suggested solution is to use a surrogate primary key, combined with three additional columns keeping the explicit time period an entry is valid and the status of every record. There can be many historical records in a database. The strategy of tracking changes is the recommended one, as it makes it possible to preserve historical data (and not change reports for past periods in turn) and reflect most recent updates at the same time.

SCD techniques - summary

SCD Type	Modeling approach	Impact
Type 0	No change	Fact associated with original value
Type 1	Overwrite	Fact associated with current value
Type 2	New row with new attribute value	Fact associated with attribute value in effect when fact occurred
Type 3	New column with prior attribute value	Fact associated with current and prior values
Type 4	Mini-dimension for rapidly changing attributes	Fact associated with rapidly changing attribute values in effect when fact occurred
Type 5	Type 4 + Type 1 mini-dimension key in base dimension	Fact associated with rapidly changing attribute values in effect when fact occurred and current rapidly changing attribute values
Type 6	Type 2 + Type 1 overwritten attributes + Type 3 prior values	Fact associated with attribute value in effect when fact occurred, plus current values
Type 7	Type 2 + view limited to current rows	Fact associated with attribute value in effect when fact occurred, plus current values

Hierarchies

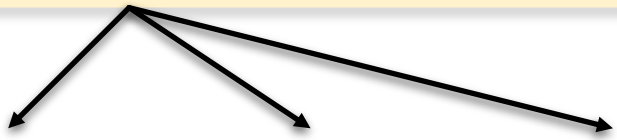
- Many dimensions rely on hierarchies:
  - Location can be decomposed from continent, down to country, region, city
  - Time can be decomposed to year, quarter, month, week and day
- Moreover, even in the same dimension table, multiple hierarchies may coexist:
  - Time can be decomposed based on:
    - Calendar year/quarter/month
    - Fiscal year/quarter/month

Hierarchy types

Hierarchy type	Data type	Example
Fixed Depth	A series of many-to-one relationship with a fixed and limited number of levels	Products are linked to product groups and these to categories
Slightly Ragged (~nierówny) / Variable Depth	A series of many-to-one relationship with a limited, but varied number of levels	Location can be decomposed to Country, Region and City, but also Country, State, Region and City. Still, the depth is limited
Ragged/Variable Depth	A series of many-to-one relationship with no clear number of levels	The organisation structure of a large enterprise will have a varied number of levels in individual subtrees (departments) with no clear limitation

Modelling hierarchies
Fixed depth case

- This is the easiest case to model
- Simply, a group of N+ attributes is added to dimension table, where N is the depth of the hierarchy



Product_id	Org_id	…	Category	Subcategory	Class	…
10450	123	…	IT	Printers	Laser printer	…
19081	123	…	IT	Printers	Ink jet printer	…
…	…	…	…	…	…	…

Please note that dictionary tables are denormalised and their content is placed in dimension table, in this example being the Product dimension table.

Modelling hierarchies
Slightly ragged case

- This is the relatively easy case to model
- Simply, a group of N+ attributes is added to dimension table, where N is the maximum depth of the hierarchy

Geo_id	Org_id	…	Country	State	Region	City	…
10450	123	…	US	New York	West	Syracause	…
19081	123	…	Poland	Poland	Mazowieckie	Warsaw	…
…	…	…		…	…	…	…

Since, maximum number of levels is supported, in some entries not all of them may be needed. In such cases, the excessive attributes (such as State in Poland) are populated with the same values as the value of parent attribute.

Modelling hierarchies
Ragged case

Parent_ organisation_key	Child_ organisation_key	Depth_ from_parent	Highest_ parent_flag	Lowest_ child_flag
1247	1568	3	FALSE	TRUE
1247	1247	0	FALSE	FALSE
…	…	…	…	…

- This is the most non-trivial case to model, as there is no clear limit on the depth of the hierarchy
- A possible real-life example is any tree-like structure such as *product structure tree* that may have a possibly large number of levels, e.g. components of a product, the parts of these components, the parts of these parts …
- In this case, [Kimball2013] suggests the use of bridge table linking fact and dimension table (here shown for organization table):

Modelling hierarchies
Ragged case

Parent_ organisation_key	Child_ organisation_key	Depth_ from_parent	Highest_ parent_flag	Lowest_ child_flag
1247	1568	3	FALSE	TRUE
1247	1247	0	FALSE	FALSE
…	…	…	…	…

- The table lists all possible combinations of (parent,child) pairs in the tree-like structure.
- For each pair, the length of the path from parent node to child note is added.
- Whether parent is the root of the hierarchy and child is a leaf is made clear.
- Parent and child organisation keys are foreign keys to the dimensions table. This is because the same value of a parent (and child key) may appear in many records in the table.
- Finally, (node,node) record is created for every node value.

Ragged case example

Order_facts table

Product_id	Order_date	…	Department	…
10450	20211220	…	1568	…
19081	20211227	…	1899	…

Bridge table

Parent_ organisation_key	Child_ organisation_key	Depth_ from_parent	Highest_ parent_flag	Lowest_ child_flag
1247	1568	3	FALSE	TRUE
1568	1568	0	FALSE	TRUE

Department_DIM table

Department_id	Department_name	…
1568	B2B Department	
1247	eCommerce Department	

A reference is made from facts table to a child key in a bridge table. It does not take the form of a foreign key, as it refers to a part of a composite PK. Additional filtering conditions are needed when joining facts with bridge table not to clone the fact data in reports.

Fact tables revisited

- In some cases, distinction between fact and dimension can be not obvious.
- Example:
  - If price tends to be constant it can be a dimension i.e. an attribute of a dimension table
  - However, if price changes frequently, then its change can be a fact
- Some facts tables can contain no measures:
  - For instance, the fact that a complaint has been made can contain no measure columns. Still, the number of such facts can be counted and may be a major performance indicator.

Problematic fact attributes

- Some fact attributes can be both measures and dimensions
- For instance, order value can be:
  - A measure, since by adding up order values we can calculate overall order value
  - A dimension, since by binning order values we can easily count the number of large orders, e.g. having the value exceeding 10 000 Euro and other orders

Treating an attribute as both a measure and a dimension is possible but should be applied carefully. Please note that list price is another attribute that can play two roles at the same time.

Aggregate fact table

- In some cases, raw facts can be too detailed i.e. many reports will require aggregates only
- Too many detailed records may slow down the analysis
- Hence, apart from transactional fact table, ***aggregate fact table*** can be developed to provide aggregated facts ready to be used

Aggregate fact can be particularly useful when underlying raw data are large and too detailed for analytical purposes. As an example, a data warehouse showing the efficiency of a smart metering system may include aggregate fact table. Such an aggregate fact table may show for which measures some data were reported for individual days. This is not to refer to individual data transmission sessions over a day each time daily data is needed.

Aggregate fact table example

Website activity tracking table in OLTP

Id	…	Session_id	Seconds_on_page	Url	Timestamp
123	…	9874	24	https://www.***	2021/12/26…

ETL process

Website activity tracking fact table in OLAP

SkId	Org_id	…	Session_id	Seconds_on_page	Url	Timestamp
102934	123	…	9874	24	https://www.***	2021/12/26…

ETL process

Website activity tracking **aggregated** fact table in OLAP

Id	…	Average_seconds_on_page	Url	Date
458	…	31	https://www.***	2021/12/26

Due to large volumes of data, raw visits can be stored also in non-OLTP platforms such as Apache Hadoop

Summary

- Dimensional modelling techniques answer the need to precisely capture the data used for analytical purposes
- This should consider several real-life constraints:
  - Identifiers of dimensions may partly overlap in underlying OLTP tables and/or change over time
  - Past data in a DWH should not be overwritten with a new data, as this would change reports for past periods
  - There is a need to model hierarchies of known and limited number of levels, varying number of levels and possible large number of levels
- Due to efficiency reasons, performing all calculations on-the-fly is not justified. Hence, techniques such as aggregate fact tables, but also date dimension tables have been developed

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zadanie 10 pn. Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych, realizowane w ramach projektu ,,NERW 2 PW. Nauka - Edukacja - Rozwój - Współpraca'', współfinansowanego ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

Business process management

Jakub Abelski, M.Sc.
J.Abelski@mini.pw.edu.pl

Hurtownie danych i systemy Business Intelligence
Wydział Matematyki i Nauk Informacyjnych
Politechnika Warszawska

Fundusze Europejskie
Wiedza Edukacja Rozwój

Rzeczpospolita Polska

Politechnika Warszawska

Unia Europejska
Europejski Fundusz Społeczny

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zadanie 10 pn. Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych, realizowane w ramach projektu ,,NERW 2 PW. Nauka - Edukacja - Rozwój - Współpraca'', współfinansowanego ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

Bibliography

- S. Mohapatra, *Business Process Reengineering. Automation Decision Points in Process Reengineering*, Springer, 2013 (ebook available at https://www.bg.pw.edu.pl)
- M. Piotrowski, *Procesy biznesowe w praktyce: projektowanie, testowanie i optymalizacja. Procesy biznesowe w polskich warunkach*, Helion, 2014 (dostępna w Bibliotece Głównej PW)
- C. Northcote Parkinson, *Parkinson's Law, and Other Studies in Administration*, Houghton Mifflin Harcourt, 1962
- A. Sharp, P. McDermott, *Workflow Modelling. Tools for Process Improvement and Application Development*, 2nd Ed., Artech House, 2009
- Resources on:
  - Business Process Management Notation (BPMN) available at www.bpmn.org
  - Business Process Execution Language (BPEL) available at www.oasis-open.org

A bit of history

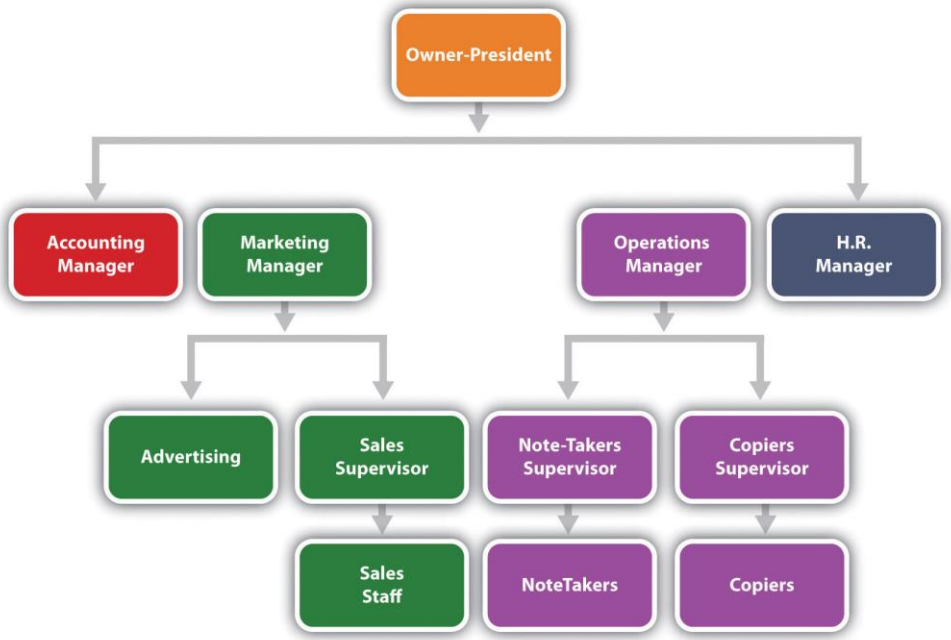
- Till XVIII century i.e. before industrial revolution:
  Skilled craftsman preparing the product from the beginning till the end i.e. incl. selling it:
  - Extensive knowledge and experience required
  - Example: blacksmith
- Pros:
  - Individuals knowing the entire process
- Cons:
  - Limited scalability: more products => more skilled craftsmen required, they should know … the entire process

Industrial revolution

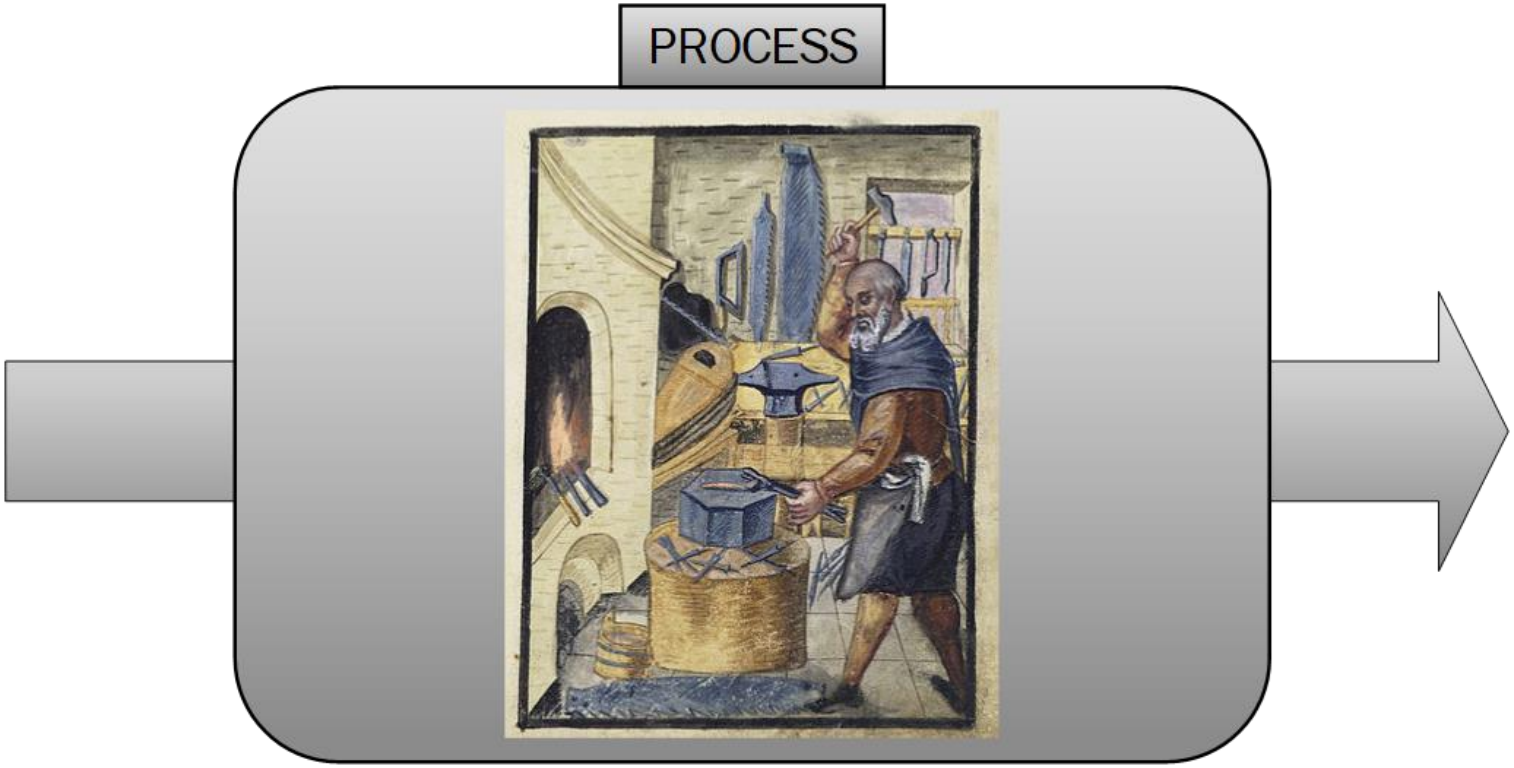
- Not just the steam engine
- Division of the labour into specialised tasks:
  - Every worker knowing and responsible just for a part of the process
  - The labour division had been soon extended to other activities: sales, accounting, …
- Pros:
  - The opportunity to create functional specialties
  - Better scalability – both up and down: extra specialists added/removed easily
  - Easier to manage a group of professionals in one field than persons spanning different skills required to perform the entire process (find clients, buy materials, produce, sell)
- Cons:
  - The big picture being an end-to-end process can be easily lost

XXth century

- Complex organisations with multiple functional departments
- Every department centred on its own functions
- A growing tendency to optimise the use of resources within a department
- This turned out to be suboptimal in the 90's:
  - What is optimal for a department is not necessarily optimal for the entire organisation and/or the client

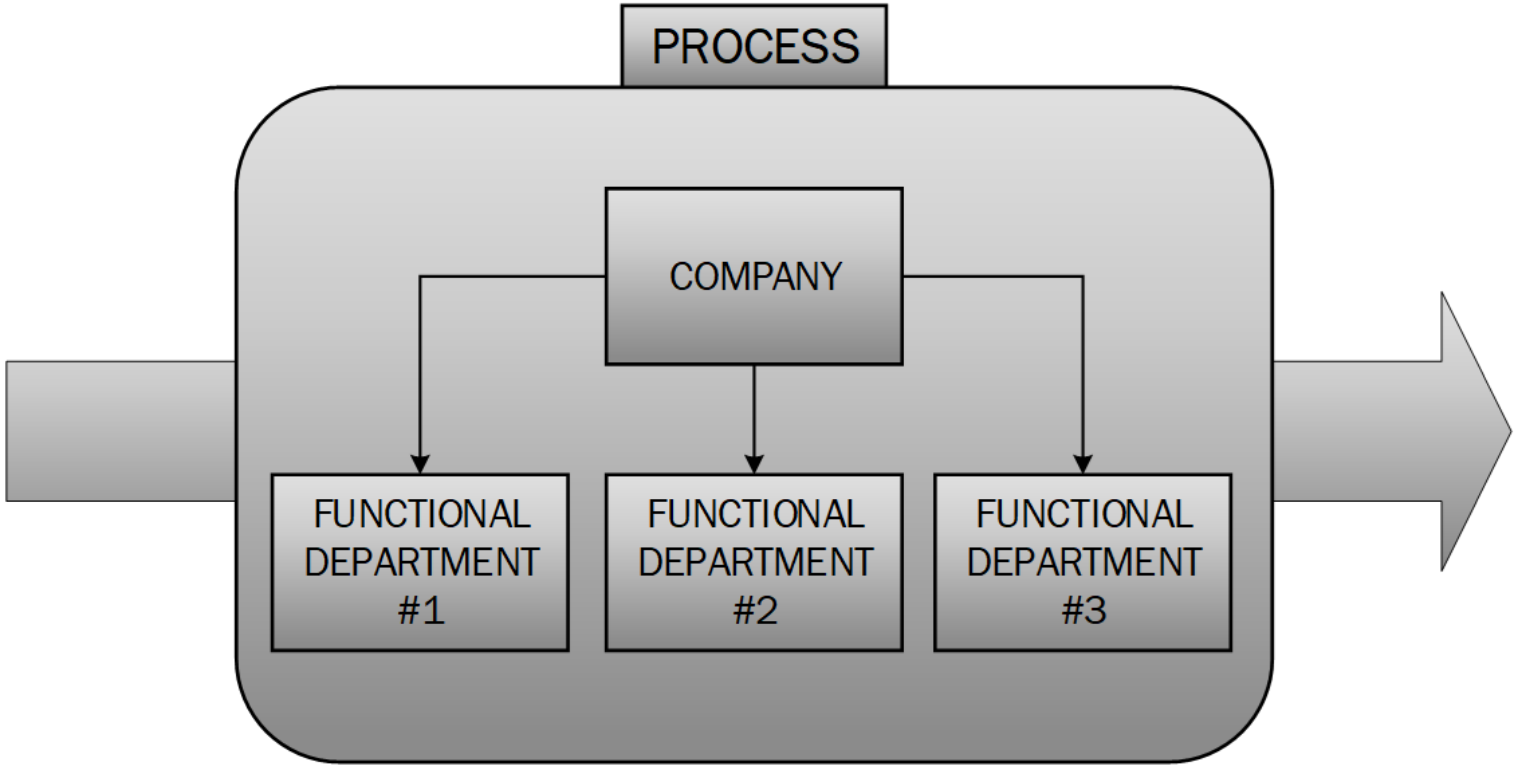


Pre-industrial model



Before industrial revolution, the craftsmen could (and had to) control the entire end-to-end process i.e. a process initiated by a client starting from negotiating the conditions of an order till the work was submitted and sold to the client.

Current model



Currently many functional departments participate in the process to deliver the value that matters for the client e.g. a product or a service. The big process can be easily lost, when none of them is/feels responsible for an entire process.

Sample process

- A client wants to order furniture for his/her kitchen made with some specific settings (materials and dimensions). The process can be done in the following steps:
  1. An order is accepted by sales office
  2. Materials are ordered by orders department
  3. Production is done in a factory
  4. Invoice is prepared by the accounting department
  5. Furniture is delivered by a shipment company.

Due to specialisation, the end-to-end process is done by many functional departments. In addition, subcontractors can be involved (here: shipment company). The needs of individual departments might be different from the needs of the client. For a factory it may make sense to group similar orders. This means delays for the clients.

The drawbacks of specialisation

- The overall process becomes fragmented, hence neither seen, nor controlled
- Functional silos can come into being i.e. departments concentrated on their work and neglecting the process
- Department-oriented optimisation prevails over customer-centric approach
- Inter-functional communication problems

The benefits of process-driven management

- The processes have to be named and described. Otherwise, the way they are done might be not clear, as it can involve many undocumented manual/spreadsheet-based steps
- The focus is placed on customer and results
- Defined processes can be measured and compared against baselines, company objectives, etc.
- Unnecessary, redundant work and rework can be identified and eliminated
- Greater awareness and participation of an entire team can be observed.

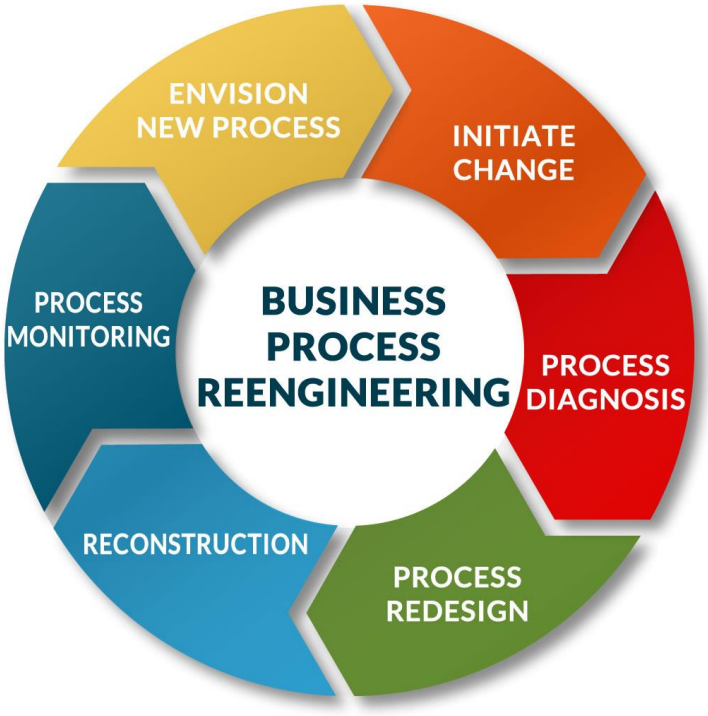
In reality, while IT systems are used for managing some areas, they are frequently accompanied by manual work, Ms Office documents, private databases, etc. To define and describe such processes means to identify the potential for improvements. In many cases this reveals problems not even known to the management staff.

Processes: the IT perspective

- Things to avoid:
  - Grouping similar tasks of a department to manage resources might be optimal for a department, but not for the client/organisation,
  - An IT system dedicated for a department with a local database might be ideally suited for a department, but not for an entire organisation
- Case study:
  - Handle customer complaints more efficiently?
  - Let us collect many malfunctioning devices before we send them to the servicing company
  - Optimal for a department, but not for the client

Business Process Reengineering

- The term coined by M. Hammer in 1990
- Got very popular in 1990-1993 period
- The idea:
  - Let us redesign existing processes to regain the "big picture" and make the processes optimal
  - This approach gained a lot of publicity initially



Business Processes nowadays

- Commonly referred to as Business Process Management (BPM) rather than BPR:
  - Reflects evolutionary rather than revolutionary approach present nowadays
  - More aware of the benefits of functional organisation in modern highly specialised world
  - Inseparable from application development activities
  - Emphasises the role of process maintenance

Business Process Reengineering used to be considered a separate set of activities done by consultants and not directly related to software engineering. For a few years, it is done as a part of one project together with business/requirement analysis. RUP business modelling discipline clearly confirms this approach.

BPM in IT projects

- Return on Investment in IT projects can be accounted to:
  - Improved business process execution
  - Better decision making
- To improve business process performance means to:
  - Define and understand business processes of an organisation
  - Use IT technology to automate processes by reducing manual work, probability of errors and rework needed

Once, process definition is changed, data warehousing technology can be used to observe the actual performance of process execution.

Business process – key aspects of a good definition

- Business process should be an end-to-end activity, not just the set of actions done by a single department
- Potentially it could be, however it means the optimization of the entire process will go beyond the scope of the project
- It should avoid ambiguity:
  - Should have a clear start and a clear end
  - Should be easily quantifiable

1. Many small processes may result in local optimisation. Hence, the need for end-to-end processes.
2. Key processes that serve external customers are frequently referred to as core processes.

Good process definition – key aspects

Aspect	Suggesion	Example
Process name	Verb-noun form Indicate process result Avoid mushy verbs	Sell Product Sign Contract Develop Sales Plan
Process result	Discrete, Countable, Identifiable, Essential	Product sold Contract signed Mail Delivered Defect repaired Product or Service or Information
Who and how does the process?	Should not affect process identification as who/how may change	Avoid: Send by mail, phone call to manager XYZ
Start of a process	A process has a triggering event	A client places an order

Examples of mushy verbs: manage, handle. Even worse process names: Customer Management, Delivery Organisation,…

Process features - discussion

Feature of a process	Description
Identifiable	Result instances can be identified. This can provide basis for KPIs. E.g. individual persons hired, products sold, defects repaired.
Countable	Identifiable and discrete results in countable. Hence KPIs can be developed.
Essential	The process is really needed for organisation / client. Counter example: print invoice. A better name: deliver invoice to the client. An invoice can be printed today to be sent via e-mail tomorrow. The important thing is to deliver the invoice. The more general name promotes the changes in the current implementation, improving business performance. By making the result essential to the client, we avoid incomplete processes showing a part of the actual client-oriented process.

Triggering event

- Three categories of process trigger events are:
  - Action event: someone makes a certain decision which starts a process e.g. a client places an order, reports a defect
  - Temporal event: happens after a predefined time, at a predefined date
  - Rule event: whenever some condition gets fulfilled, the event takes place and triggers the process

Process workflow

- Defined as:
  Work plan for responding to a triggering event. It shows the sequence of steps, decisions, and handoffs carried out by the process's actors between the initial event and the final result (Sharp, McDermott, 2009)
- Actor can be:
  - A person
  - An organization
  - An IT system
  - Some machinery

Workflow Management System is an IT system that allows to define and execute workflows of different processes in an organisation. It automates the processes, which may reduce the cost of processes comparing to manual paper-based workflow.

Process area

- Collections of processes can be identified to create process areas
- These can be named with mushy verbs e.g. Customer Management process area
- Still, individual processes should avoid mushy verbs

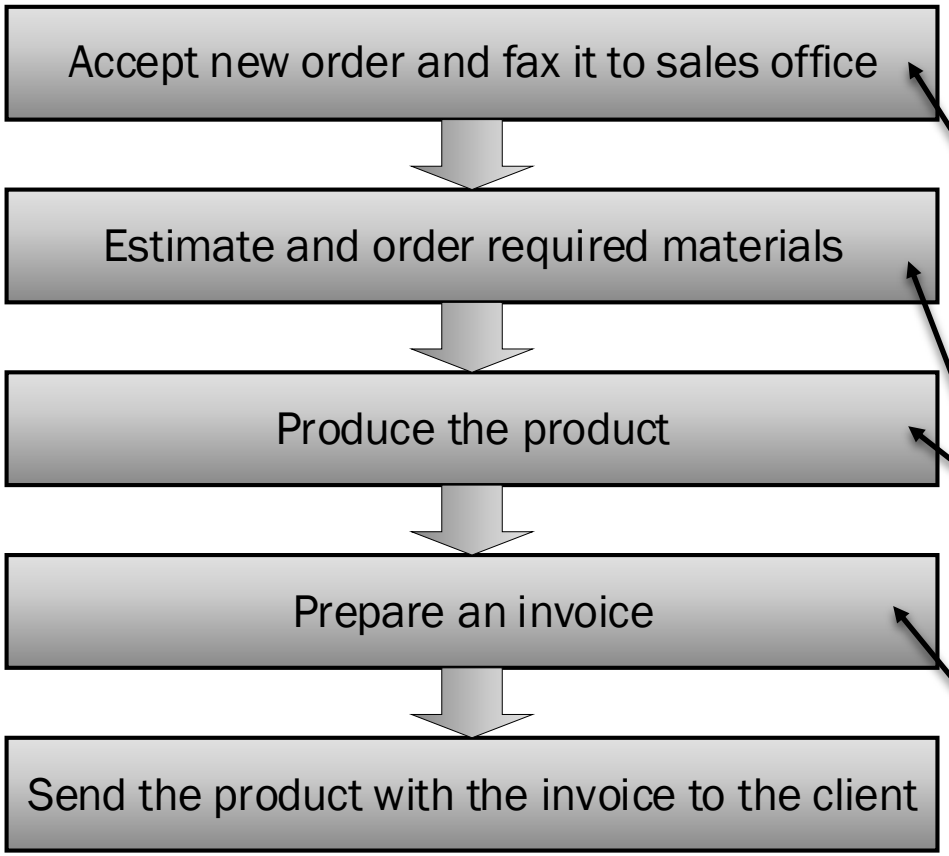
A common mistake is to try to create a single process out of a process area. Hence, multiple triggering events and process ends can be observed, which makes process identification, description and automation far more complicated, if possible.

Process – steps and activities

Process definition	Used when	Examples
A sequence of steps	When a standard a priori defined algorithm is used e.g. to deliver invoices	Deliver Invoice
A set of activities	Used when no a priori sequence of steps can be identified, as due to creativity needs this should be avoided.	Promote New Brand

Strictly repetitive processes can be defined with a sequence of steps. On the other hand, more creative processes can not assume a fixed sequence of steps. For instance, depending on the scale of a marketing campaign and the competitors, different activities can be considered. In the latter case, a process is defined with a set of possible activities rather than the sequence of steps.

Case study:
Deliver Ordered Furniture Process

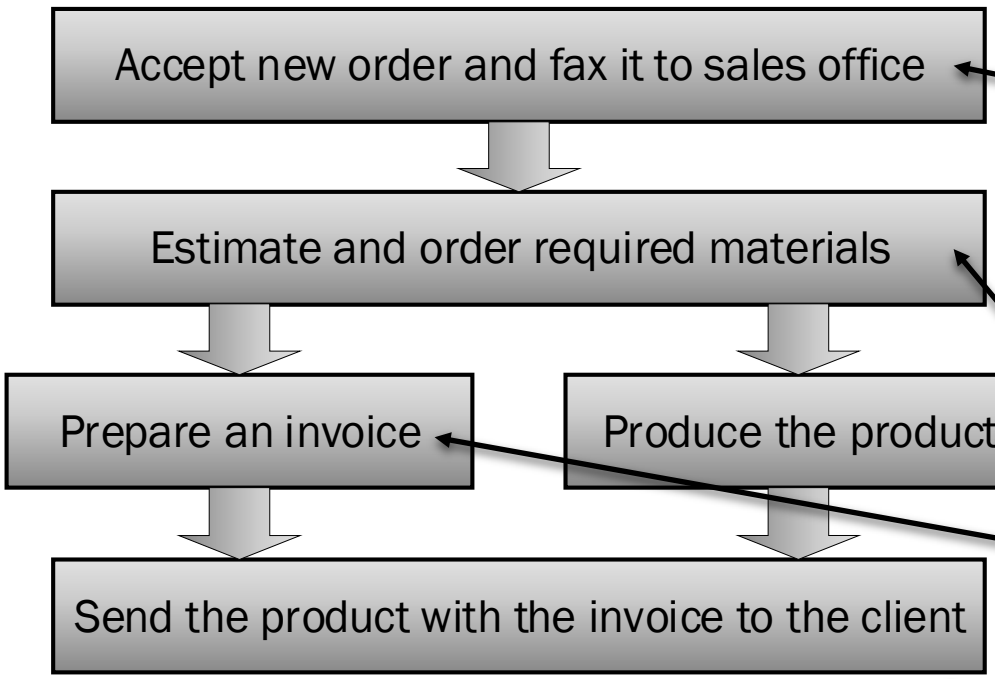


A client comes to the shopping centre. The order is written on paper and faxed to the central office, to be typed into the system.
Problems:
1. inaccurate orders; relevant data missing,
2. materials no longer available offered by the sales representatives and present in the orders
Hence: delays, unsatisfied customers

Done in a production system

Done in an accounting system, by the accounting department based on the printouts from production system.
Problems: some mistakes now and then.
The client is not billed for what he/she has ordered

Case study:
Deliver Ordered Furniture Process - reengineered



The sales representatives present in the shopping centres, use web interface to production systems for collecting the orders. Hence, incomplete orders missing some details and/or referring to materials no longer available are eliminated. This means that delays and extra work needed to resolve problematic orders are eliminated.

Done in a production system

Done in an accounting system, by the accounting department. The order data is supplied to the accounting system automatically.

Managed by the IT system of the shipment company. Delivery status requires phone calls.

Business process documentation vs data modelling

- Business processes are key sources of organization data stored in a various formats, e.g. database records.
- When a new business area is planned to be integrated into analytical platform, the documentation of business processes becomes one of the most important inputs to determine the datasets to be ingested and transformed, as well as to define potential reporting needs.
- The data discovery phase is a joint effort of business analysts, data architects, and business SMEs, usually including the following steps:
  - Identification of business processes, relevant diagrams and definitions
  - Preparation of data flow / data journey diagrams
  - Preparation of conceptual data models
  - Filling in the inventory of data focusing on datasets description and measures logic.

Case study
The use of accumulating snapshot fact table

Delivery_fact table

Order_ id	Submit_ date	Materials_ ordered_ date	Submit_to_ materials_ ordered_lag	Ready_for_ delivery_ date	Submit_to_ ready_lag	…
101	2021/12/10	2021/12/12	2	2021/12/20	10	…
102	2021/12/10	2021/12/10	0	2021/12/22	12	…
…	…	…	…	…	…	…
110	202			2021		

For every process execution (here: delivery of every order) we track performance of individual stages

We can easily find out that for some orders, it took us too much time to perform some steps

This example revisits the idea of accumulating snapshot table. A DW table is used to track the performance of process execution. This can largely help to observe whether the reengineered process is more efficient than the original one. In addition, the performance of every step/task can be easily tracked.

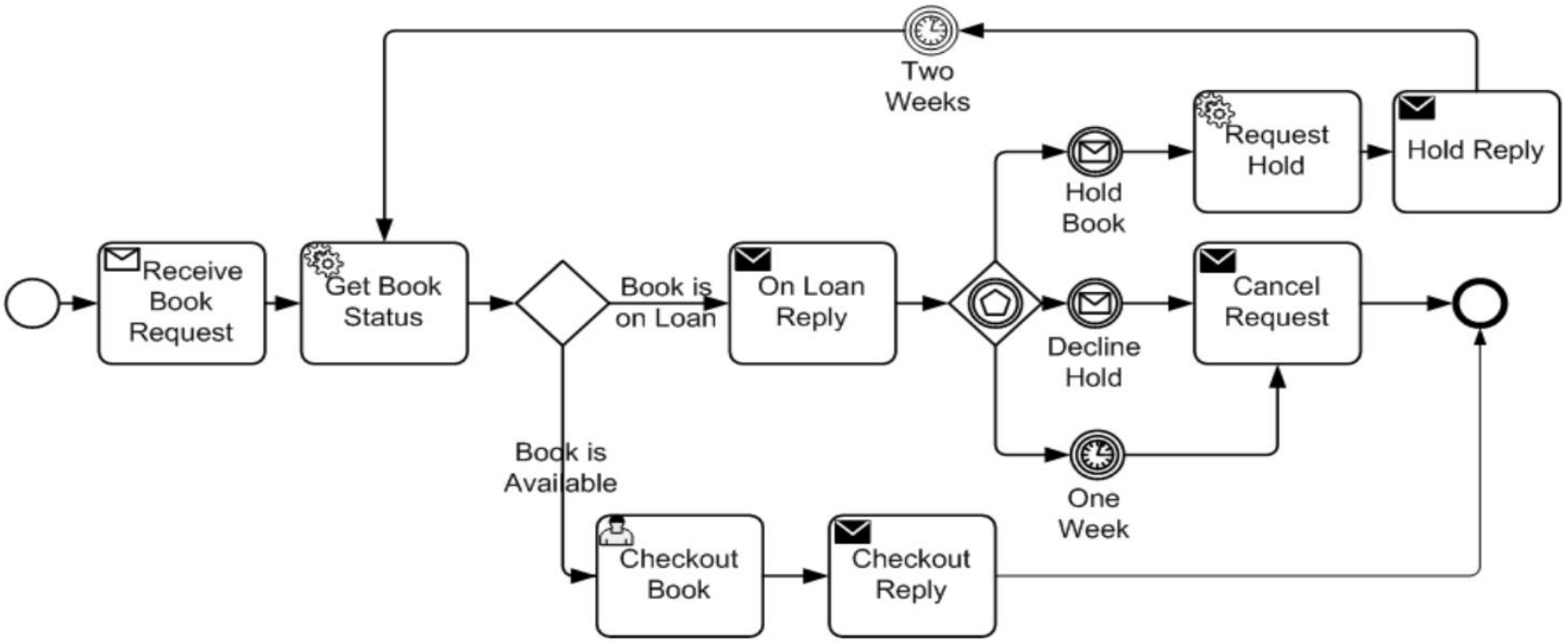
BP Reengineering – some considerations

- To improve BP execution usually means to (partly) automate them
- This means IT systems have to be involved, integrated and/or redesigned to answer new requirements
- On the other hand, an IT project to meet business goals has to modify existing BP, to:
  - Eliminate unnecessary work
  - Increase parallelism
  - Eliminate delays due to manual processing and inevitable errors caused by it
- By assessing the cost of rework and/or delays eliminated, ROI for the project can be estimated

Related IT standards

Standard	Meaning	Description
BPMN	Business Process Management Notation	A notation standard for modelling business processes. Details available at http://www.bpmn.org/ The standard is developed by Object Management Group. The processes can go beyond IT steps and include manual processing, too.
BPEL	Business Process Execution Language	An XML-based language for defining business processes. The execution steps can be done by calling web services exposed by participating system. The standard is available at http://www.oasis-open.org/ and is developed by Organization for the Advancement of Structured Information Standards

BPMN process – sample diagram



Many actions are time-based, which emphasizes the role of workflow management system.
Image source: http://www.omg.org/spec/BPMN/2.0/PDF/

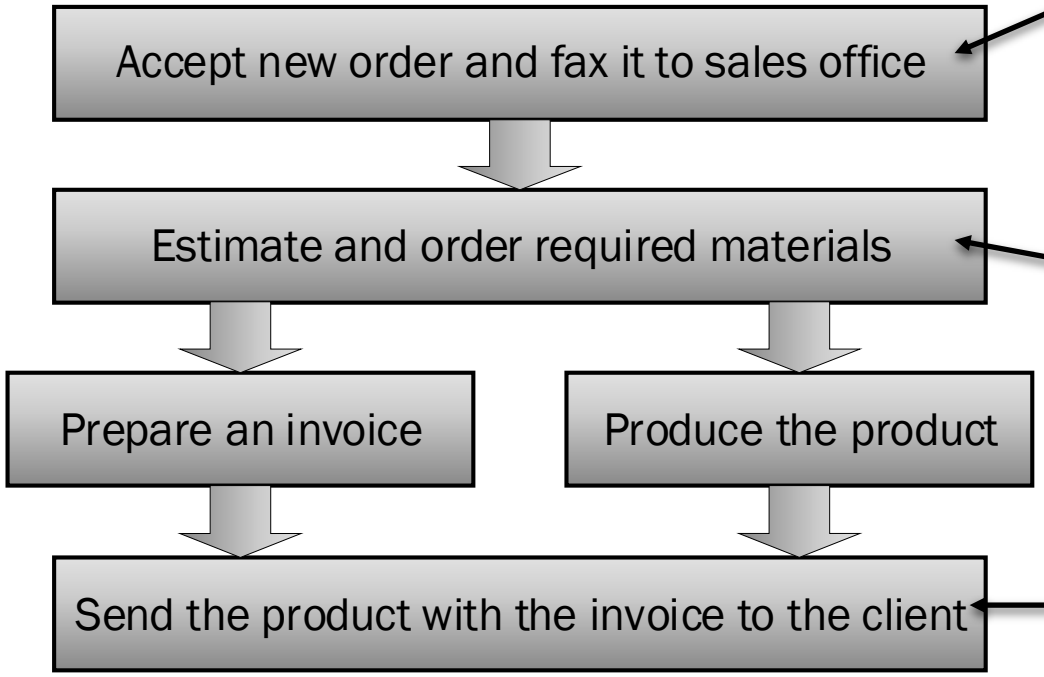
Business Process automation

- Can be done easily within one IT system
- Becomes more complicated when multiple IT systems have to participate:
  - Some of them might be not available
  - Instead of RPC-oriented web services, message-based loose integration might be preferred
- The process automation becomes even more problematic, when some of the steps are done manually e.g. some decisions of management staff are required for the process to continue

In such cases, special middleware products can be used to run and monitor processes. In fact, the process execution may no longer "belong" to any participating system.

Case study:

Deliver Ordered Furniture Process – the possible use of process engine



The process instance is initiated. From now on, the business process engine will monitor the execution and initiate next steps.

Once this step is finished, the engine will create an invoice (by calling a web service of acc. system) and initiate the production (by calling a web service of prod. system).

1. The engine finds out that an invoice has been created and the product has been marked as finished.
2. It calls the web service of a shipment company (if possible) or creates a manual task to inform the company about new shipment.

Summary

- Business processes can and should be identified
- Continuous improvement of a business process is vital for every organisation
- Whether new process definitions actually yield performance improvements they can and should be monitored with data warehousing solutions
- Hence, to design a data warehouse for an enterprise means to focus on business processes of the enterprise

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zadanie 10 pn. Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych, realizowane w ramach projektu ,,NERW 2 PW. Nauka - Edukacja - Rozwój - Współpraca'', współfinansowanego ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

Enterprise Data Warehouse Bus Architecture

Jakub Abelski, M.Sc.
J.Abelski@mini.pw.edu.pl

Hurtownie danych i systemy Business Intelligence
Wydział Matematyki i Nauk Informacyjnych
Politechnika Warszawska

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

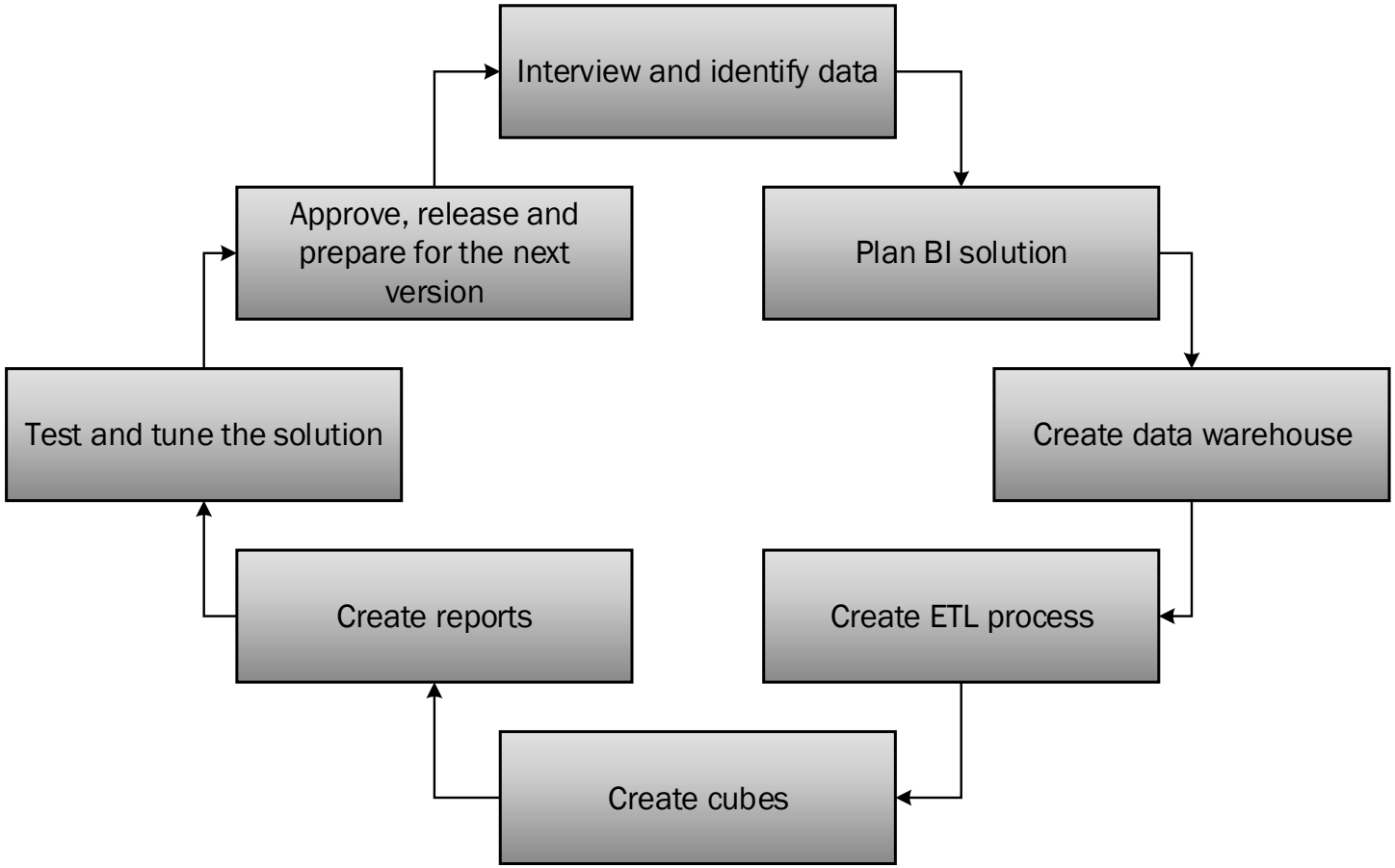
Zadanie 10 pn. Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych, realizowane w ramach projektu ,,NERW 2 PW. Nauka - Edukacja - Rozwój - Współpraca'', współfinansowanego ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

Key references

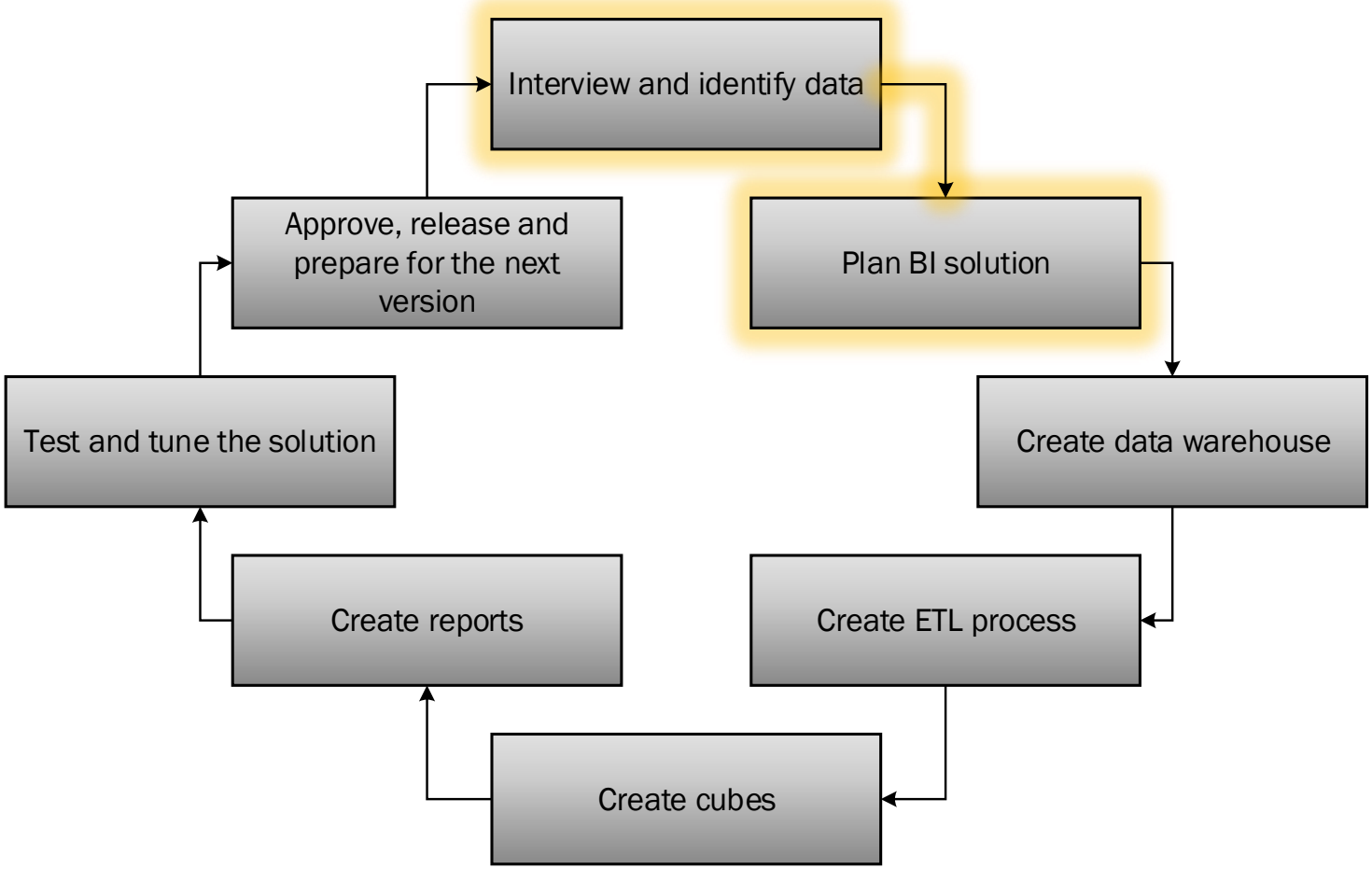
- [**Kimball2016**] Kimball, Ralph and Ross, Margy, *The Kimball Group Reader: Relentlessly Practical Tools for Data Warehousing and Business Intelligence*, Second Edition, Wiley, 2016 (available as an e-book via Main Library portal available at https://bg.pw.edu.pl/)
- [**Simon2014**] Simon, Alan, *Enterprise Business Intelligence and Data Warehousing: Program Management Essentials*, Morgan Kaufmann, 2014 (available as an e-book via Main Library portal available at https://bg.pw.edu.pl/)
- [**Howson2014**], C. Howson, *Successful Business Intelligence, Second Edition: Unlock the Value of BI & Big Data*, McGrow Hill Education, 2013
- [**Kimball2013**] Kimball, R., Ross, M., *The Data Warehouse Toolkit. The Definitive Guide to Dimensional Modelling*, Wiley, 3rd Ed., 2013

BI solution life cycle - reminder



Even before BI solution can be planned, data has to be investigated first. Equally importantly, creating a data warehouse combined with ETL process is not a final step, since front-end part i.e. interactive data presentation based on Business Intelligence system is needed.

BI solution life cycle – planning stage



One of the approaches to plan BI solution is to use Enterprise Data Warehouse bus architecture introduced by R. Kimball and described in [Kimball2013]

Enterprise Data Warehouse architecture

- Proposed by R. Kimball in 1990s to decompose the process of designing a DW/BI solution
- Independent from DW/BI software platform
- Emphasises the need for business benefits as a driving factor behind any successful DW/BI initiative
- Promotes development of a DW/BI solution for an entire organisation and reuse of the same dimensions
- Relies on the following key artefacts:
  - Enterprise Data Warehouse Bus Matrix
  - Opportunity/Stakeholder Matrix

Enterprise Data Warehouse Bus Matrix

- A key proposal of Enterprise Data Warehouse Bus architecture
- Defined by:
  - **Rows** being business processes to be supported by a DW/BI solution
  - **Columns** being dimensions
- Frequently takes the form of a square matrix of ca. 25-50 columns and rows
- May require substantial (but needed) work and support of higher management to overcome silo-tendencies and solutions potentially existing in various departments
- In addition, a more granular bus matrix addressing the detailed implementation can be defined. Here, each business process row is expanded to show specific fact tables or OLAP cubes and the precise data grain.

Enterprise Data Warehouse Bus Matrix example

Process\ Dimension	Delivery date dimension	Vendor dimension	Product dimension	Shipment method dimension	…
Deliver product	X		X	X	
Buy materials		X	X		
Destroy expired product		X	X		
…					

Development of the matrix inevitably involves stakeholders from different departments and promotes dimension re-use. For instance, the process owner of Buy materials proces may observe that he/she would like to have the process described by shipment method (e.g. regular mail, own transport, vendor transport, …) to analyse the number of partly damaged materials received by the procurement department.

Opportunity/Stakeholder Matrix

Process/ Stakeholder group	Sales representatives	Procurement	Inventory	Help desk	…
Deliver product	X		X	X	
Buy materials		X	X		
Destroy expired product			X		
…					

This matrix is used to map processes to key stakeholders. This is to clarify which stakeholders are involved in the development of requirements for individual processes. It can also help declaring the need to access individual data.

Design guidelines

- Development of Enterprise Data Warehouse Bus Matrix promotes:
  - The refinement and use of the same dimensions across entire organisation
  - This should result in the same ways of processing the data and defining dimensions
- Processes are the rows, not departments or other organisation units. Hence, client-oriented perspective is promoted (as long as correct business process definitions exist)
- The risk of generalisation of "similar" dimensions should be avoided. Hence, salesperson, product engineer or crew manager should be the dimensions to place in the matrix, rather than a generalised "person" dimension

Conformed dimensions (*wymiary uzgodnione*)

- Conformed dimensions are descriptive master reference data that is referenced in multiple dimensional models and re-used across different fact tables
- In every table they are used in, they have the same:
  - Attribute names
  - Domain content (or a subset of the domain content of another dimension in the group of conformed dimensions)
- Conformed dimensions are also referred to as: master dimensions, common dimensions, reference dimensions and shared dimensions
- Conformed dimensions are a fundamental element of the Kimball data warehouse modeling approach

The benefits of conformed dimensions

- Conformed dimensions reduce development cost, as they:
  - Can be handled once within the ETL process - allow data from different sources to be integrated based on common, unified dimension attributes
  - Allow a dimension table to be built and maintained once rather than recreating slightly different versions during each development cycle
  - Promote the use of the same names and concepts throughout an organisation
- Conformed dimensions can be used for drill across operation i.e. for integrating different fact tables sharing the conformed dimension. They allow DW/BI users to consistently slice-and-dice performance metrics from multiple business process data sources

Different variants of conformed dimensions

- Conformed dimensions can be:
  - Identical for all business areas, i.e. have the same number of rows, key values, attribute labels and values
  - Shrunken dimensions of two categories:
    - Shrunken with row subset for instance, e.g. only some of the employees are product managers
    - Shrunken with attribute subset e.g. country dimension used in one fact table can be an attribute-based subset of a dimension including city and country present in another fact table
- Shrunken dimensions can be created in multiple ways:
  - Creation of the base dimension containing lowest level data and then creation of a shrunken dimension from the base dimension
  - Creation of the shrunken and base dimensions separately – when there are separate source tables for the shrunken dimension attributes. Shrunken dimension can be joined back to the base dimension early in the ETL process to populate the current higher-level attributes

Summary

- Creating a data warehouse spanning entire organisation is a challenge
- Importantly, a correct data warehouse should be directly related to core business processes of the organisation
- Enterprise Data Warehouse provides a structured approach to develop data warehouse:
  - Focused on key business processes
  - Involving relevant stakeholders
  - Identifying conformed dimensions, i.e. dimensions shared across departments and stakeholder groups

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zadanie 10 pn. Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych, realizowane w ramach projektu ,,NERW 2 PW. Nauka - Edukacja - Rozwój - Współpraca'', współfinansowanego ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

Data integration - ETL / ELT processes

Jakub Abelski, M.Sc.
J.Abelski@mini.pw.edu.pl

Hurtownie danych i systemy Business Intelligence
Wydział Matematyki i Nauk Informacyjnych
Politechnika Warszawska

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zadanie 10 pn. Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych, realizowane w ramach projektu ,,NERW 2 PW. Nauka - Edukacja - Rozwój - Współpraca'', współfinansowanego ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

References

- [**KimbalETL**] Kimbal R.,*The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*, 1st Edition, Wiley, 2004
- [**ReeveETL**] Reeve A., *Managing Data in Motion: Data Integration Best Practice Techniques and Technologies*, 1st Edition, Morgan Kaufmann, 2013
- [**DycheETL**] Dyche J., Levy E., *Customer Data Integration: Reaching a Single Version of the Truth (SAS Institute Inc.)*, 1st Edition, Wiley, 2006

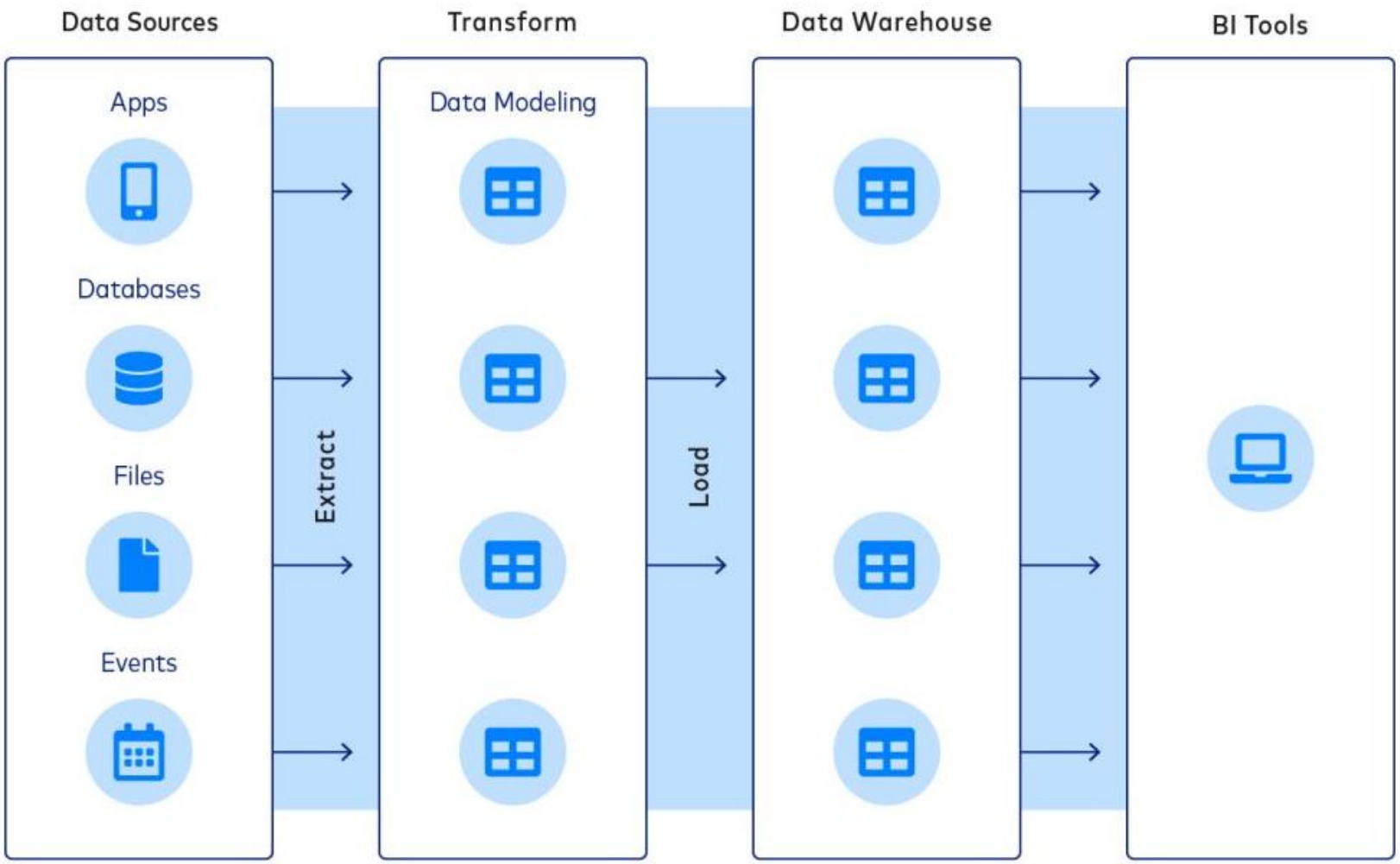
Data Integration

- ***Data Integration*** *comprises the practices, architectural techniques and tools for achieving the consistent access and delivery of data across the spectrum of data subject areas and data structure types in the enterprise to meet the data consumption requirements of all applications and business processes* (Gartner)
- Example types of data integration techniques:
  - ETL/ELT batch data integration – process in batches at scheduled time
  - Stream data integration – real-time data load through data pipelines
  - Application integration – the use of API to move and synchronize data between applications
  - Data virtualization – uniform access through a virtual layer on top of data sources (real-time on demand)
  - Change data capture – identifying and capturing changes in a source system and delivering those changes in real-time to a target system
  - Data consolidation – staging of raw data and consolidation into a single view (cleansing, standardization, harmonization)

Extract-Transform-Load (ETL)

- ETL is an automated process including:
  - **Extraction** of information required for analysis from a raw data source, usually OLTP database, but also data files or web streams. Often, multiple and different types of sources are collected on this stage to be combined for further needs.
  - **Transformation** of extracted data from its original form into the well-defined target structure so that it can be loaded into another database. Transformation often relies on pre-defined rules and lookup tables and standardizes multiple data sources to allow storage of results in single location.
  - **Loading** of data into the target database (usually data warehouse)
- ETL typically summarizes data to reduce its size and improve performance for specific types of analysis
- Building an ETL infrastructure requires integration of various data sources and thorough testing to ensure data is transformed correctly and no crucial information is lost

ETL data flow example



Source: https://www.fivetran.com/blog/how-to-compare-etl-tools

Data integration tools – ETL processes

- Example popular tools for implementing ETL processes:
  - **Informatica PowerCenter** - a mature, feature-rich enterprise data integration platform for ETL workloads
  - **IBM Infosphere Information Center -** ETL product from IBM designed mainly for Big Data companies and large-scale enterprises
  - **Oracle Data Integrator** - a comprehensive data integration solution that is part of Oracle's ecosystem
  - **SQL Server Integration Services (SSIS)** - ETL product dedicated to Microsoft SQL Server
  - **Talend** - open-source ETL data integration solution
  - **BusinessObjects Data Integrator** - data integration ETL tool for SAP platform
  - **SAS Data Integration Studio** - GUI to build and manage data integration processes on SAS platform
  - **Apache Nifi** - Big Data oriented, scalable directed graphs of data routing, transformation, and system mediation logic
  - **Fivetran** – ELT data movement platform for cloud solutions

Magic Quadrant for Data Integration Tools (22-23)

- Magic Quadrant (MQ) is a series of market research reports prepared by Gartner company. They are built based on proprietary qualitative data analysis methods to visualise market trends including participants, their position, directions and maturity.
- The data integration leaders according to this report are Informatica, Oracle and IBM.

Figure 1: Magic Quadrant for Data Integration Tools

The drawbacks of in-house ETL solutions

- Data integration tools are widely used in enterprise architectures and strongly promoted by solution architects
- Building of a custom, in-house system might look like an interesting option to reduce the overall cost, however it:
  - Requires a variety of IT competences (back-end, front-end, networking, databases, administration) and support for a wide range of data endpoints (database products, FTP servers, APIs)
  - Requires building a rich user interface to define, configure and monitor the ETL processes
  - Is associated with additional maintenance cost related to software upgrades and bug fixing
  - Should be constantly updated and monitored for the security compliance
- Therefore, usually these responsibilities are transferred to the vendor of ETL platform

Implementation recommendations – part I

- **Data extraction from various sources**
  - The basis for the success of subsequent ETL steps is to extract data correctly. Most ETL systems combine data from multiple sources, each with its own data organization and format – including relational databases, non-relational databases, XML, JSON, CSV files, etc. When extracting the data, it is recommended to convert files into a single format for standardized processing.
  - All necessary data sources should be discovered and described with a proper technical (data access strategy) and business (types of available data) considerations.
  - Source data is usually described in the data inventory catalog and profiled to investigate data types, statistics, null options, etc.
- **Reference data**
  - Create a set of data dictionaries defining a list of permissible values.
  - For example, in a country data field, a list of country codes can be specified.

Implementation recommendations – part II

- **Data validation**
  - Data validation should be an automated process that confirms whether data pulled from sources is complete and correct.
  - For example, in a database of financial transactions from the past year, a date field should contain valid dates for the past 12 months.
  - The validation engine rejects data, if it fails the validation rules.
  - The rejected records should be available for further analysis to identify problems and address them in the data extraction process.
- **Data transformation**
  - Removing of extraneous or erroneous data (cleaning)
  - Applying business rules to data
  - Checking data integrity (ensuring that the data was not corrupted in source, or corrupted by ETL, and that no data was dropped in previous stages)
  - Creating data aggregates if necessary
  - The set of transformation rules should be clearly defined in a source-to-target mapping documentation (STTM)

STTM Examples

TARGET						SOURCE					
Mapping Change Date	Target Table	Target Column	Nullable	PK	Data-type, Length	Source Table	Source Column	Data-type, Length	Expression, Transformation	Default Value	Error Types and Handling
Y/M/D	Name of target table	Target table column name	Whether a field can be null	Primary key field for target	Data type & length for target column	Name of source table	Column in source table from which data is extracted	Data type and length for this source column	Decodes, aggregates, conversions, if statements, lookup functions	Value to use in target field when source field is null	Used to document Not null, value if looked up, pk, fk, etc…comments, issues

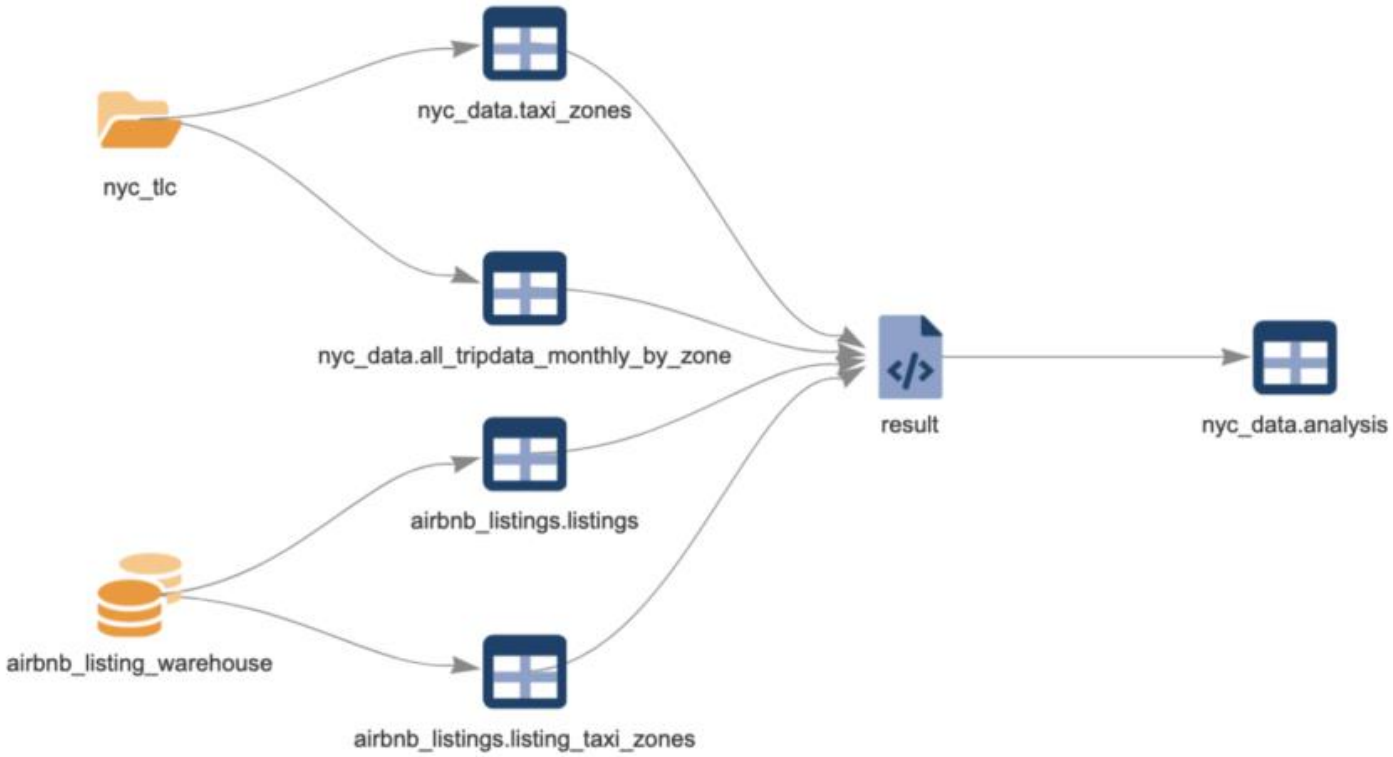
ETL Package Name	*Load_Dim_Product.dtsx*
Target Table	*ADDW.Dim.Product*
Source Tables	*Production.Product* *Production.ProductSubcategory*
Load Frequency	*Daily*
Target Type	*SCD Type 1*
Mapping Description	AD.Production.Product is the lowest level of source data. Join the source tables in the following manner: FROM AD.Production.Product INNER JOIN AD.Production.ProductSubcategory ON Product.SubcategoryID = ProductSubcategory.SubcategoryID INNER JOIN AD.Production.ProductCategory ON ProductSubcategory.ProductCategoryID = ProductCategory.ProductCategoryID
Error Handling	On an error occuring any inserted records or updates needs to roll back and leave the system in the previous correct state.

Source: https://www.ewsolutions.com/the-importance-of-data-mapping-for-data-integration-projects/

Implementation recommendations – part III

- **Staging**
  - Input data should be inserted to a staging database/tables first, making it easier to roll back in case of issues.
  - Additional audit reports can be defined on this level – to diagnose and repair potential data quality and completeness problems.
- **Data publishing**
  - Loading the data to the target tables (usually partitions) according to the pre-defined data refresh calendar (daily, weekly, monthly, …) or based on a trigger (e.g. a new file loaded to the input directory).
  - As part of data lineage documentation, new data should be extended with some technical columns including load timestamp, source system identifier, etc.
- **Logging and auditing**
  - Logging of activities that occur before, during, and after the ETL process and checking of data completeness on each processing step.
  - A load without errors is not necessarily a successful load. The ETL system should support auditing of different metrics like count of rows, sum of financial measures, etc.

Data Lineage Example



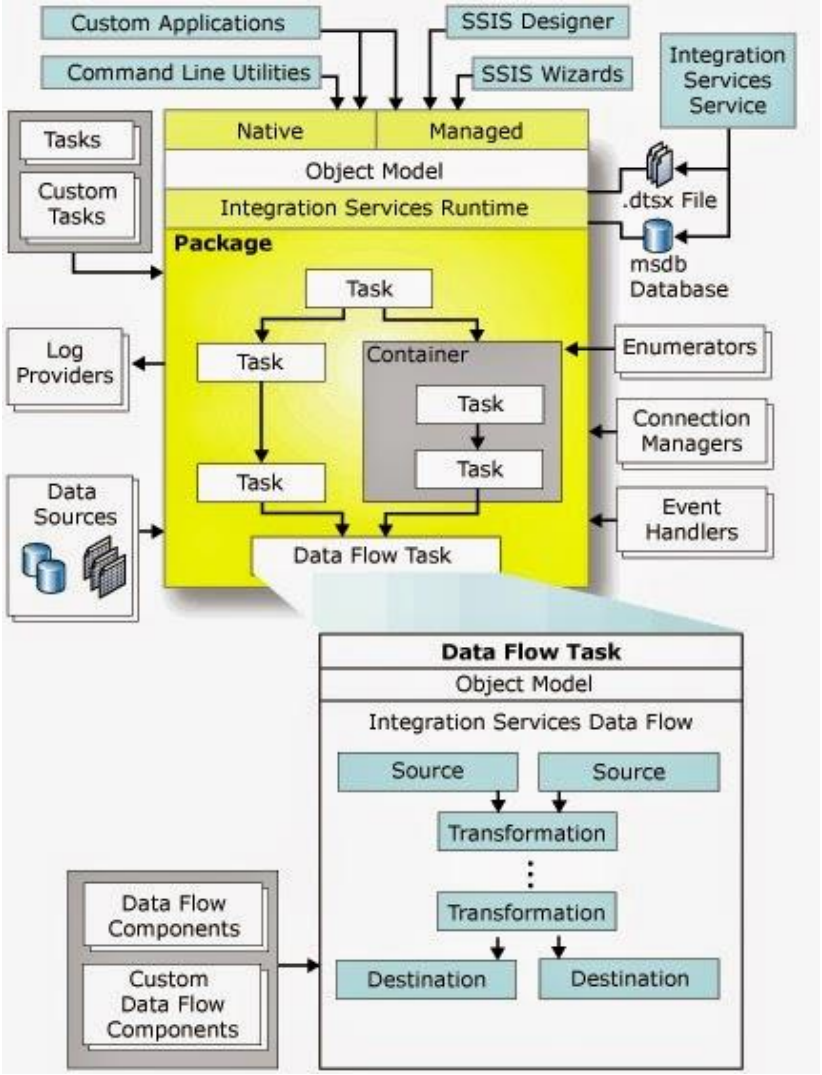
Source: https://www.silect.is/blog/know-your-data-lineage/

Example ETL solution - SSIS

- **SQL Server Integration Services (SSIS)** is a platform for building enterprise-level data integration and data transformations solutions.
- It is a component of the Microsoft SQL Server database software that can be used to execute a wide range of data migration tasks.
- SSIS:
  - Is a fast & flexible data integration tool used for data extraction, loading and transformation (cleaning, aggregating, merging data, etc.)
  - Can extract data from a wide range of sources like SQL Server databases, Excel files, Oracle and DB2 databases
  - Includes graphical tools & wizards for performing workflow functions like sending email messages, FTP operations, data sources, and destinations
  - Includes Catalog database to store, run, and manage packages

SSIS architecture - overview of the complexity

- Popular ETL solutions have a complex architecture including multiple components and processing layers
- For example, SSIS consists of the following types of elements:
  - Package
  - Container
  - Task
  - Data Flow
  - Event Handler
  - Connection Manager
- Image source: https://bageshkumarbagi-msbi.blogspot.com/2014/12/ssis-architecture-sql-server.html



SSIS components – part I

- **Package**
  - Organized collection of tasks and constraints to manage and execute tasks in an order
  - Compiled in an XML structured file
- **Control Flow**
  - Consists of one or more tasks and containers that executes when package runs
  - Orchestrates the order of execution for all its components
- **Container**
  - Used for grouping tasks together into units of work
  - Supports repeating control flows in packages
  - Three main types of containers in SSIS are:
    - **Sequence Container**: groups the package into multiple separate control flows, each containing one or more tasks and containers.
    - **For Loop Container**: defines a repeating control flow in a package (FOR)
    - **Foreach Loop Container**: defines a repeating control flow in a package (FOREACH)

SSIS components – part II

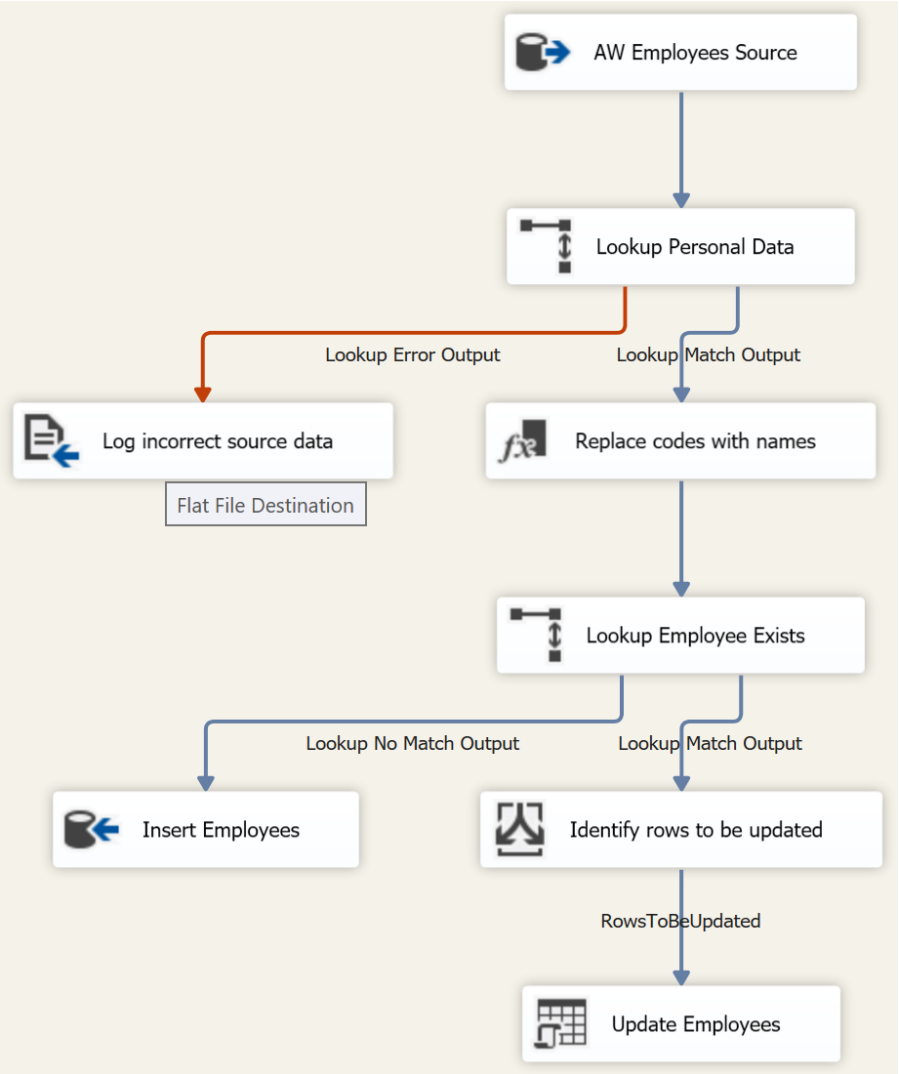
- **Task**
  - Control flow element that define a unit of work
- **Precedence Constraint**
  - Links executables, containers, and tasks in packages in a control flow
  - Specifies conditions that determine whether executables should run
- **Data Flow**
  - Task that organizes the ETL process by defining sources, transformations, and destinations
  - Example elements: ADO NET Source, Oracle Source, Derived Column Transformation, Pivot Transformation, Flat File Destination
- **Connection Manager**
  - Named connection string to data source
- **Parameter**
  - Allows to assign values to properties within packages at the time of package execution

Example SSIS tasks

Task Name	Descriptions
Execute SQL Task	Execute SQL statement in a relational database.
Data Flow Task	This task can read data from one or more sources. Transform the data when it is in the memory and write it out against one or more destinations.
Execute Package Task	Execute a package within the same project.
File System Task	Perform manipulations in the file system (create/modify/delete files or directories).
FTP Tasks	Allows to perform basic FTP functionalities.
Script Task	This is a blank task. Enables preparation of custom code within .NET framework
Send Mail Task	Allow to send an email to notify users about processing completion or errors.
Bulk Insert Task	Loading of data into a table by using the bulk insert command.
Web Service Task	It executes a method on a web service.

Example data flow

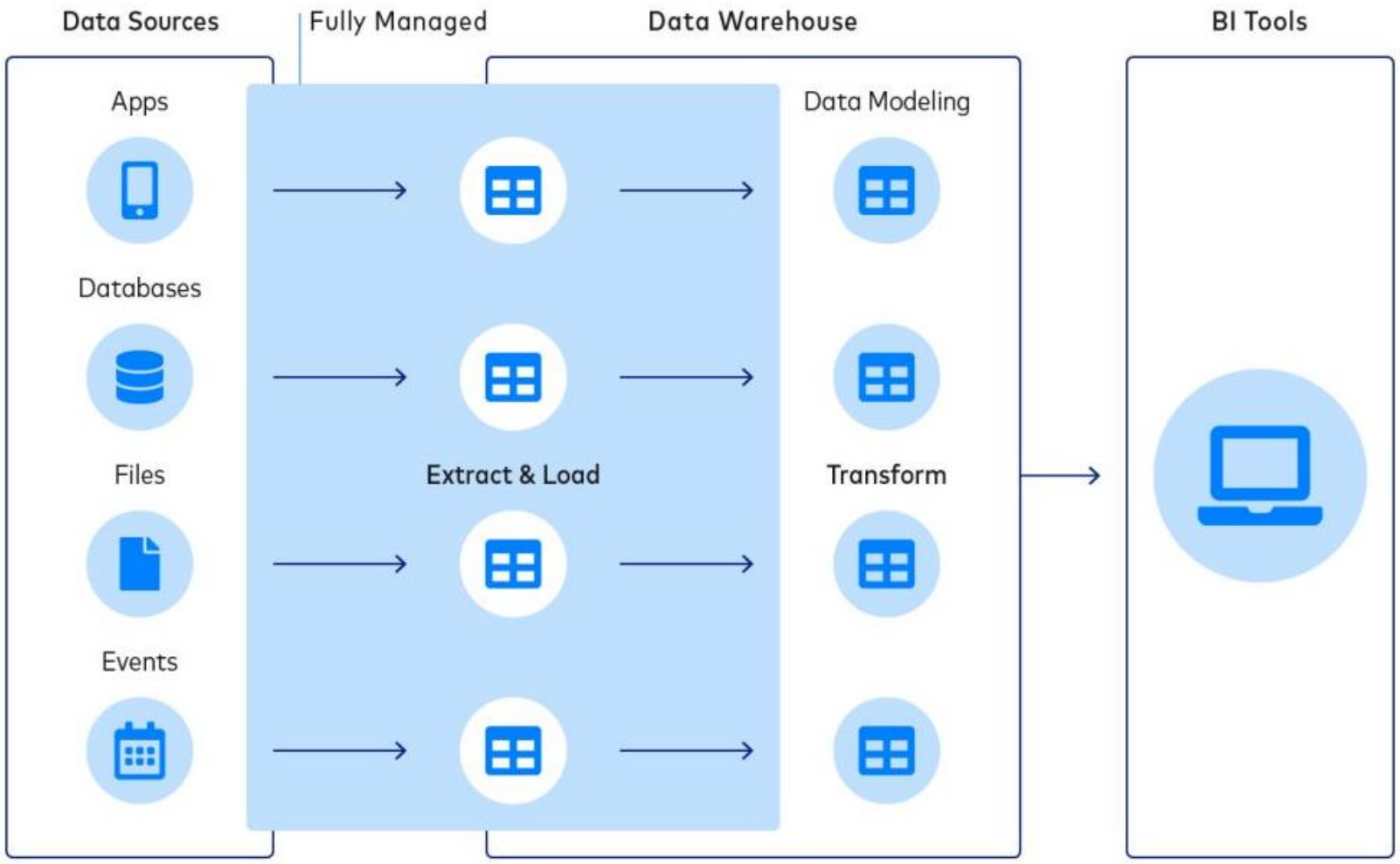
- Load of data to Employee dimension (SCD#1 upsert):
  - Access source OLTP table
  - Lookup personal data from another table
  - Replacing codes with meaningful names (possibly through reference dictionaries)
  - Lookup the current version of DimEmployee to identify rows to update / insert
  - Insert new rows to DimEmployee
  - Update rows in DimEmployee
- Alternatively, a staging table for intermediate data storage can be used



Extract-Load-Transform (ELT)

- ELT is a data integration process transferring raw data from a source location to a target system (etc. data warehouse, data lake) and then transforming the information for downstream use-cases
- In the ELT process:
  - Data is extracted in a raw format from data sources
  - Data is immediately moved into a centralized repository
  - Later, the data is transformed according to business needs
- ELT leverages the data warehouse compute power to perform data transformations
- There is no need for intermediate data staging
- ELT relies on two main rules:
  - Ingest everything to make the data available
  - Transform only the data you need

ELT – typical implementation



Source: https://www.fivetran.com/blog/how-to-compare-etl-tools

ETL vs ELT comparison

	ETL	ELT
Adoption	Well developed proces used for over 20 years	Relatively new technology, still evolving
Data availability	Transforms and loads the data that meets the data warehouse needs, only part of data will be available	Load all data immediately, later determine which data to transform and analyze
Calculations	Calculations can replace existing columns or extend the original dataset	Caculated columns can be added directly to existing dataset
Compliance	Can redact and remove sensitive information before putting it into the data warehouse or cloud server	Requires to upload the data before redacting/removing sensitive information
Data size	Smaller data sets that require complex transformations	Massive amounts of structured and unstructured data
DW support	Requires relational or structured data format	Supports structured, semi-structured, unstructured and raw data formats

Summary

- ETL is a process that extracts, transforms, and loads data from multiple sources into a data warehouse or other unified data repository
- ETL tools serve a wide range of data processing features:
  - Comprehensive automation for different data flows
  - Support for complex data management and building of consistency
  - Data quality validation, monitoring and auditing
  - Drag-and-drop interface to easily define the integration flows
  - Security of processed data and compliance of a legal aspects
- A modern variation of the ETL is ELT where the data loading take place before the data transformation.
- ETL and ELT are not the only methods for data integration. Other concepts like change data capture, data virtualization or federation can be considered as well.

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zadanie 10 pn. Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych, realizowane w ramach projektu ,,NERW 2 PW. Nauka - Edukacja - Rozwój - Współpraca'', współfinansowanego ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

Business Intelligence systems

Jakub Abelski, M.Sc.
J.Abelski@mini.pw.edu.pl

Hurtownie danych i systemy Business Intelligence
Wydział Matematyki i Nauk Informacyjnych
Politechnika Warszawska

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zadanie 10 pn. Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych, realizowane w ramach projektu ,,NERW 2 PW. Nauka - Edukacja - Rozwój - Współpraca'', współfinansowanego ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

References

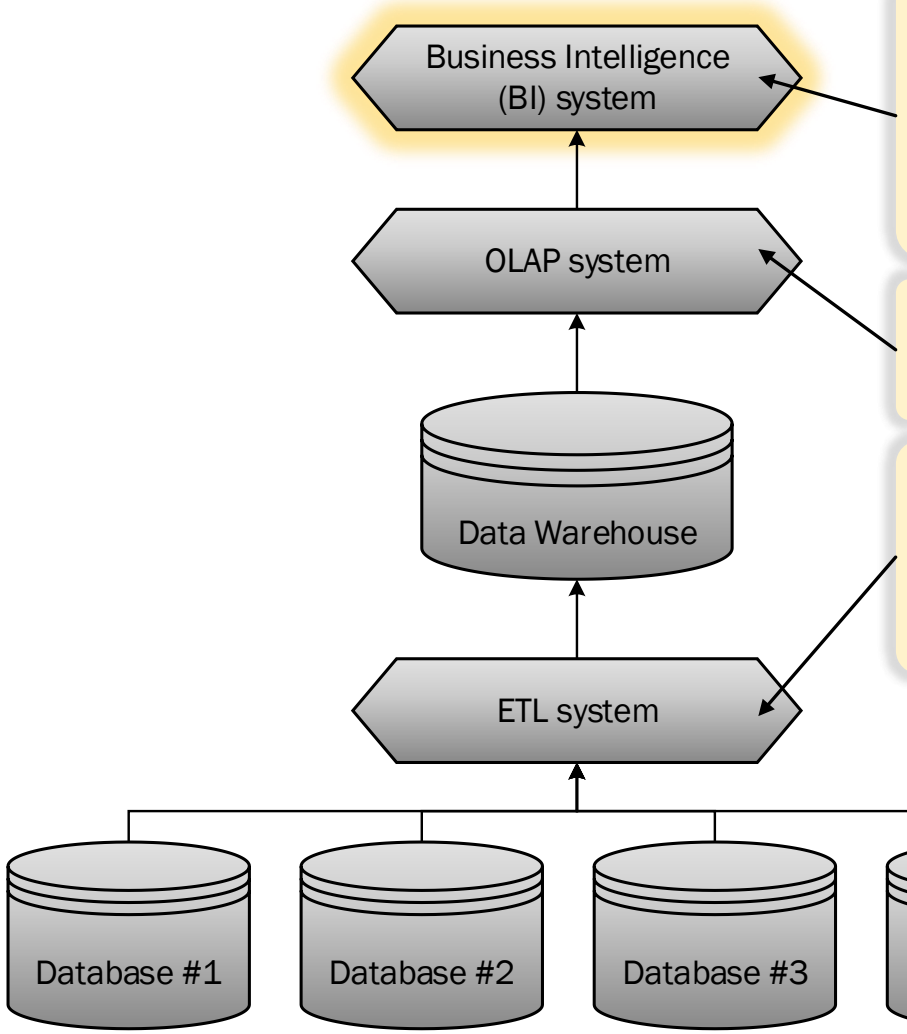
- **[Rittman2013]** Mark Rittman, *Oracle Business Intelligence 11g Developers Guide*, Oracle Press, 2013
- **[Ward2017]** Adrian Ward, Christian Screen, Haroun Khan, *Oracle Business Intelligence Enterprise Edition 12c*, Second Edition, Packt Publishing, 2017
- **[Oracle2021]** *Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition,* Oracle, 2021 (documentation available at: https://docs.oracle.com/middleware/bi12214/biee/BIEMG/toc.htm)

Business Intelligence revisited

- To have a data warehouse **does not** mean to have a BI.
- *"A key part of BI deployment are the tools that let users transform data into useful information"* [Howson2014, Root2012]. This relies on efficient DW, but also reporting mechanisms.
- Hence, BI is not a synonym of a data warehouse, even though it relies on data warehouses in most implementations

1. A popular way of providing reporting capabilities is to make them available via web applications and/or plug-ins for office applications such as Microsoft Office
2. Web-based user interface is an interesting option, as it enables access to BI software, with limited Total Cost of Ownership (TCO).

DWH/BI architecture summary



BI tools are expected to be used primarily by business staff to navigate through the data warehouse and/or analyse the data. Include reporting, querying, data analysis and data visualization tools

Database systems that primarily involve aggregating large amounts of data in a DWH

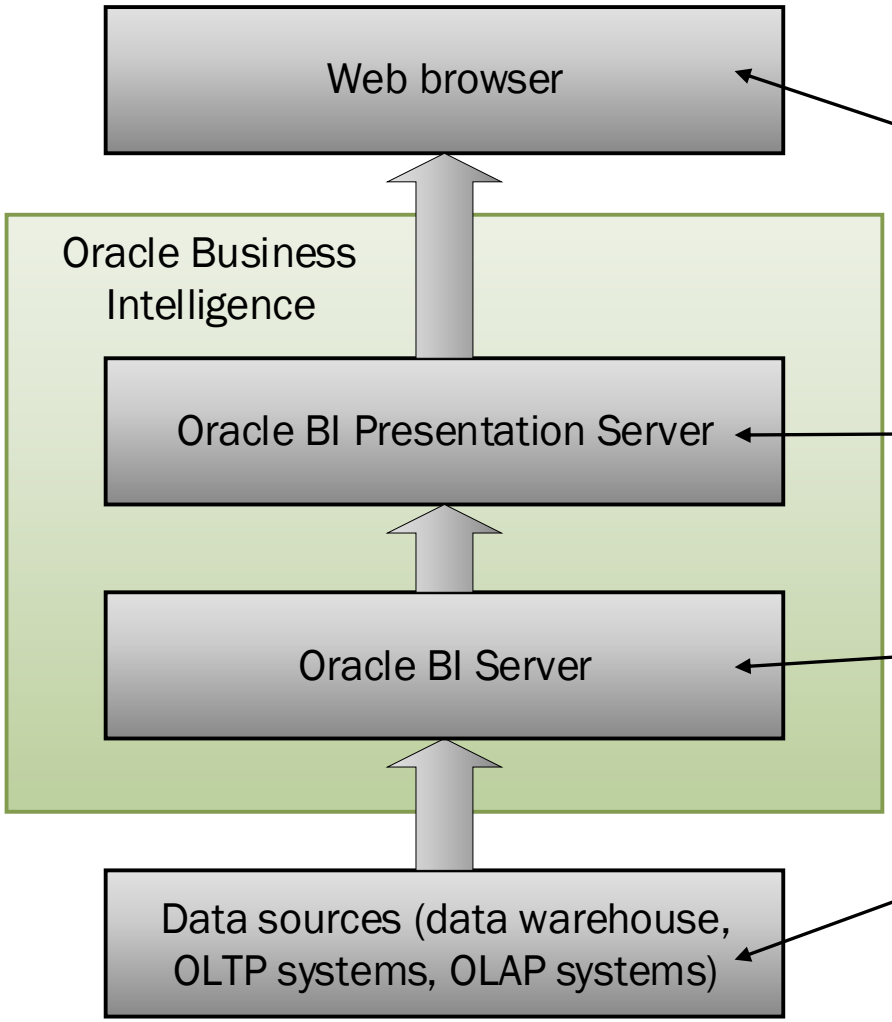
Systems used to Extract data from operational sources, Transform data for import into the data warehouse, and Load the transformed data into the data warehouse

This is where source data is placed in by operational teams such as salespersons, accounting staff and everyone using OLTP system to support his/her everyday operations

Oracle Business Intelligence

- One of the most popular BI software suites
- Available in the form of a web application Hence, a web browser is used to:
  - Design reports, dashboards and figures
  - Work with reports, dashboards and figures
- Makes it possible to analyse and visualise data from variety of data sources including non-Oracle databases
- Analytics products:
  - **Oracle Business Intelligence 12c** – enterprise on-premise BI platform introduced in 2015 (newest version released in October 2019)
  - **Oracle Analytics Cloud** and **Oracle Analytics Server** – successors of OBIEE 12c (first release: November 2019, latest release: March 2023)
  - **Oracle Analytics Desktop** – standalone data exploration and visualization
  - **Fusion Analytics Warehouse** – real-time access to personalized business application benchmarks

Oracle Business Intelligence – user perspective



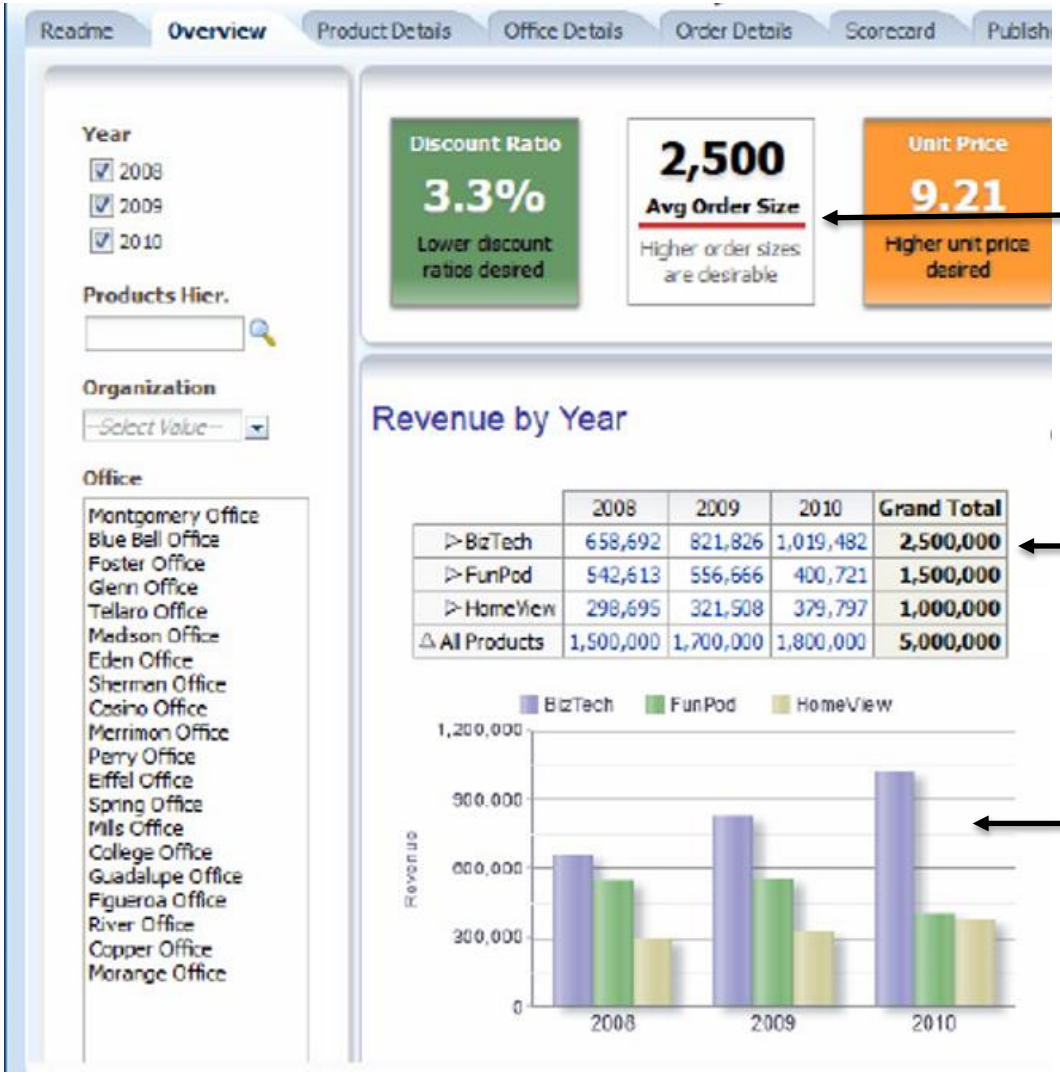
Used by end users including IT users and business users to access dashboards, reports and other analytical artefacts

This server connects to Oracle BI Server to handle dashboards, analyses and reports

This server provides simultaneous access to various data sources.

Data sources of different categories can be used by Oracle BI

Visualisation capabilities



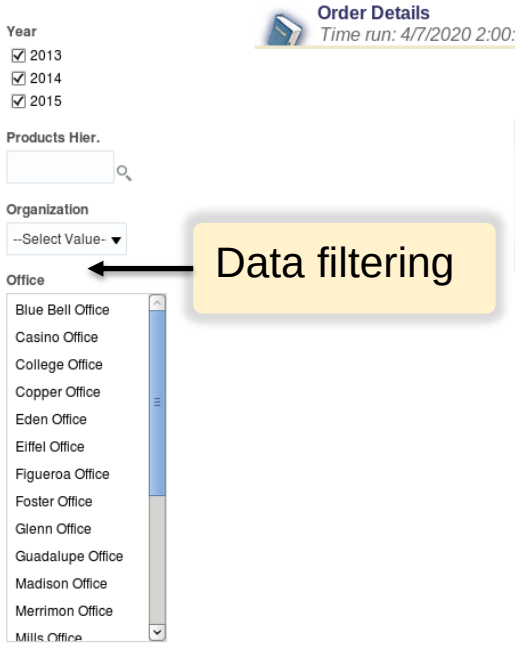
Dashboard with a number of tabs

Interactive tables with drill-down capabilities

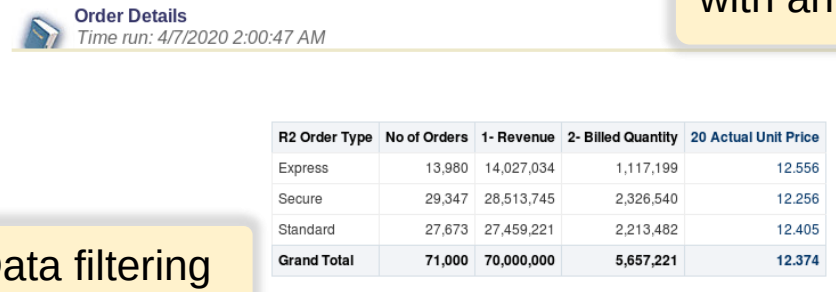
Interactive figures with drill-down capabilities

Visualisation capabilities

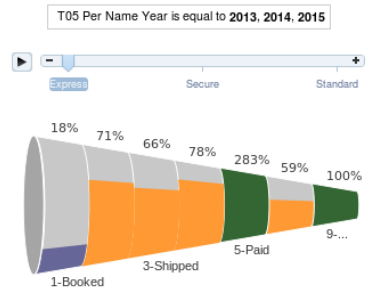
Complex visualizations with animation capabilities



Data filtering



R2 Order Type	No of Orders	1- Revenue	2- Billed Quantity	20 Actual Unit Price
Express	13,980	14,027,034	1,117,199	12.556
Secure	29,347	28,513,745	2,326,540	12.256
Standard	27,673	27,459,221	2,213,482	12.405
Grand Total	**71,000**	**70,000,000**	**5,657,221**	**12.374**

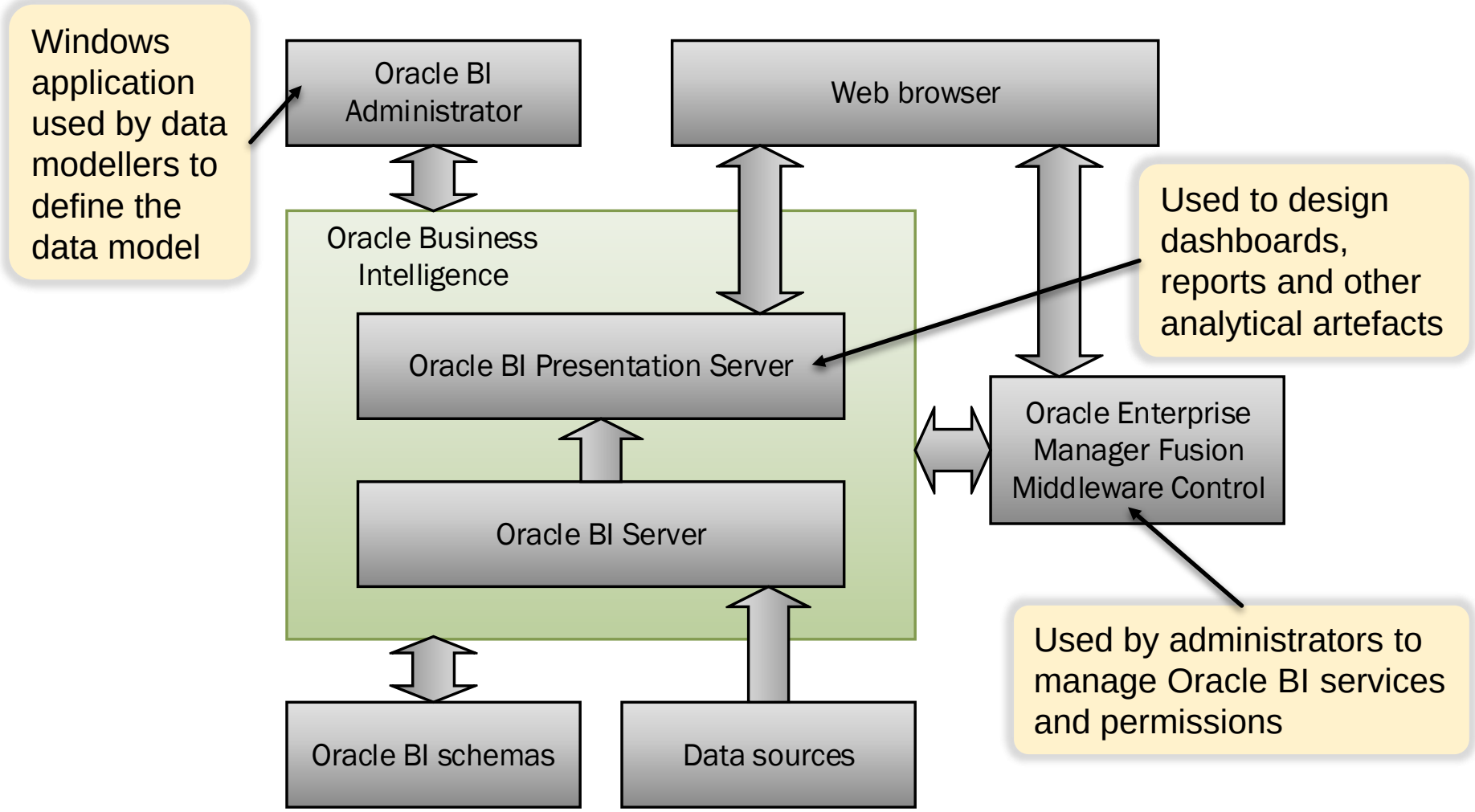


R2 Order Type All Values

Order Number	R2 Order Type	R11 Order Date-Time	R1 Order Status	P1 Product	D4 Company	1- Revenue	2- Billed Quantity
1	Secure	12/21/2012 12:00:00 AM	1-Booked	MaxiFun 2000	Stockplus Inc.	1,121	109
2	Standard	12/22/2012 12:00:00 AM	3-Shipped	MPEG4 Camcorder	Genmind Corp	487	29
3	Secure	12/18/2012 12:00:00 AM	3-Shipped	Bluetooth Adaptor	Genmind Corp	320	51
4	Standard	12/15/2012 12:00:00 AM	2-Fulfilled	MaxiFun 2000	Tescare Ltd.	105	11
5	Secure	12/15/2012 12:00:00 AM	2-Fulfilled	Tungsten E Plasma TV	Genmind Corp	100	5
6	Secure	12/21/2012 12:00:00 AM	4-Billed	V5x Flip Phone	Tescare Ltd.	427	37
7	Express	12/23/2012 12:00:00 AM	4-Billed	Game Station	Tescare Ltd.	380	18
8	Secure	12/21/2012 12:00:00 AM	2-Fulfilled	Maintenance	Stockplus Inc.	269	40
9	Standard	12/21/2012 12:00:00 AM	6-Cancelled	PocketFun ES	Genmind Corp	479	58

Links to correlated dashboards

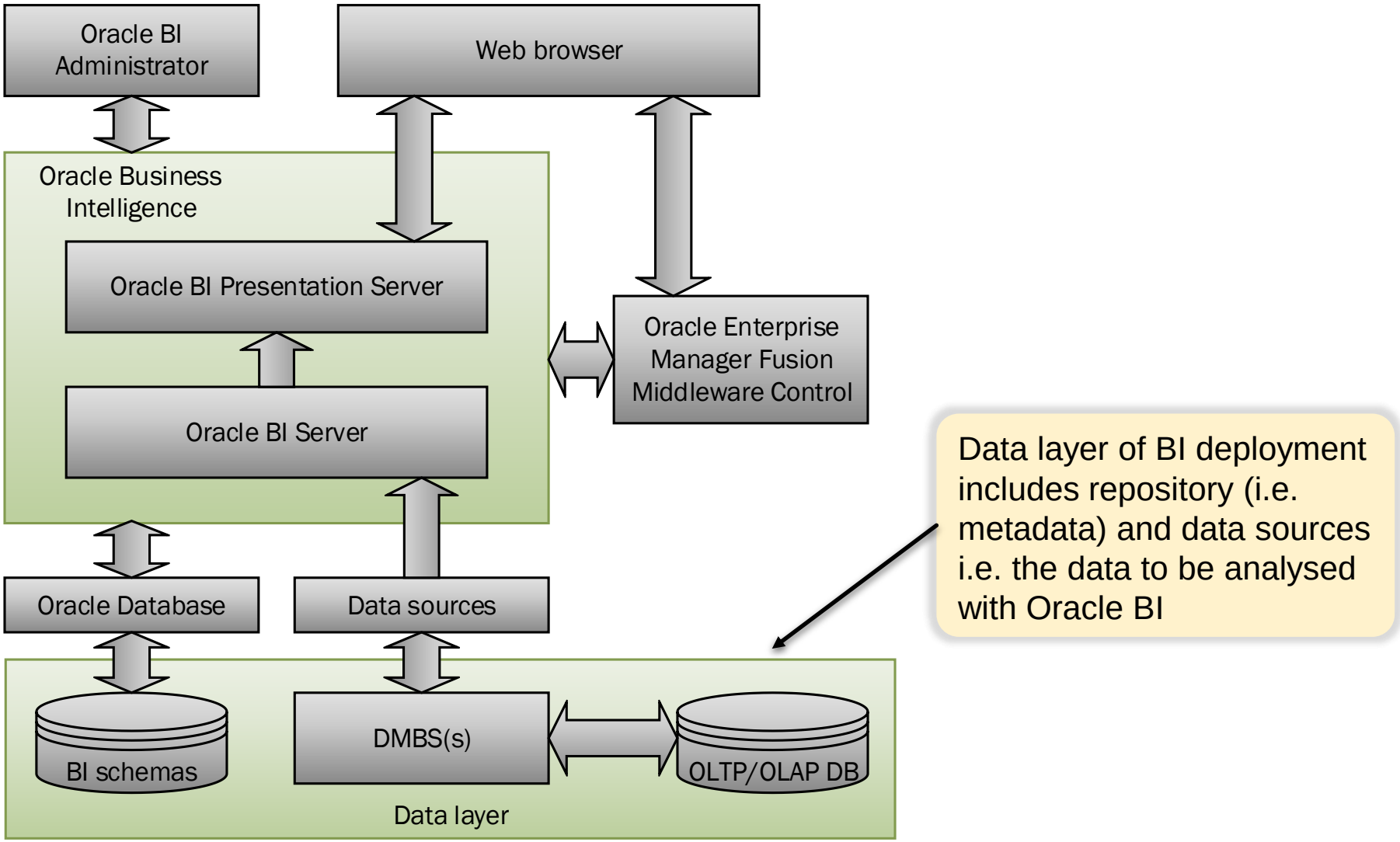
Oracle Business Intelligence – development perspective



Oracle Business Intelligence schemas

- Before Oracle BI can be installed, the schemas for its metadata have to be created
- These schemas are created with Repository Creation Utility (RCU). It creates two schemas in a database:
  - Metadata Services (MDS)
  - Business Intelligence Platform (BIPLATFORM)
- Supported databases (for schemas, not data sources) are:
  - Oracle Database
  - Microsoft SQL Server
  - IBM DB2
- Hence, a frequent choice is to have Oracle BI schemas created in Oracle database

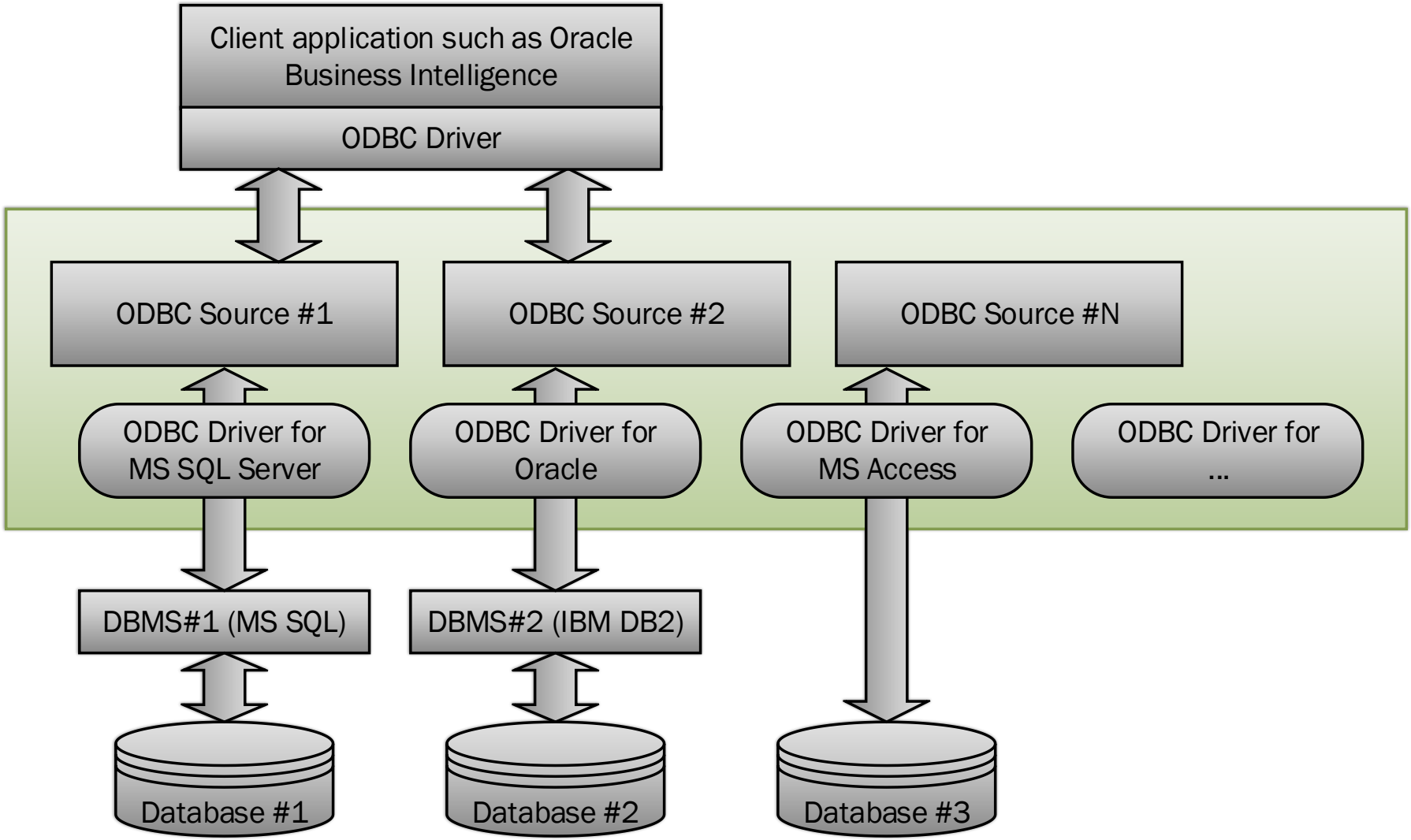
Oracle Business Intelligence development perspective - frequent layout



ODBC

- Microsoft Open Database Connectivity (ODBC) is a programming language interface that makes it possible for various applications to access databases and other sources such as MS Excel files
- The advantage of ODBC is that client applications can access variety of data stores in the same way, by exploiting ODBC drivers existing in the environment of their operating system
- The ODBC standard is strongly linked to MS Windows being its native platform. Still, ODBC-compliant components exist for non-Microsoft systems.
- Oracle BI uses ODBC to access some of the data sources, for which no dedicated Oracle BI components exist (including MS SQL Server).
- Further details on ODBC: https://docs.microsoft.com/en-us/sql/odbc/microsoft-open-database-connectivity-odbc

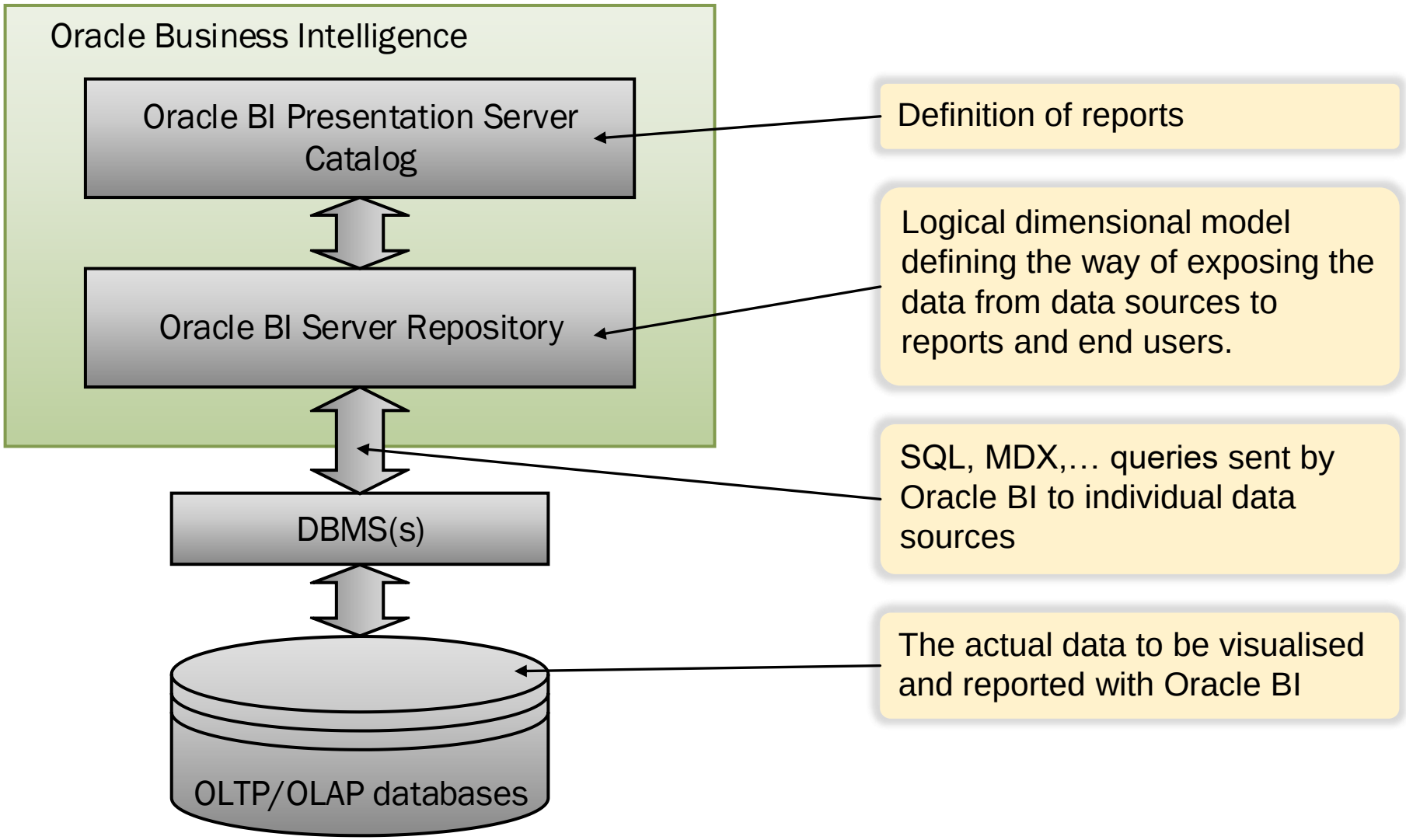
ODBC-based database connections



Oracle BI repository

- Oracle BI repository is a metadata store that contains the logical dimensional models:
  - These logical dimensional models are used by Oracle BI users to create their reports.
  - Oracle BI users do not access data stores such as data warehouses or operational OLTP databases directly. Instead, they use the extra layer of Oracle BI repository.
- The repository is mostly used by Oracle BI Server as a basis to translate the queries arising from BI user needs to SQL or MDX queries sent to physical data sources such as Oracle Database or Ms MSQL Server databases.
- Hence, both OLTP databases (queried with SQL) and OLAP databases queried with Multidimensional Expressions (MDX) query language are supported as possible data sources

Three layers of data and metadata in Oracle BI

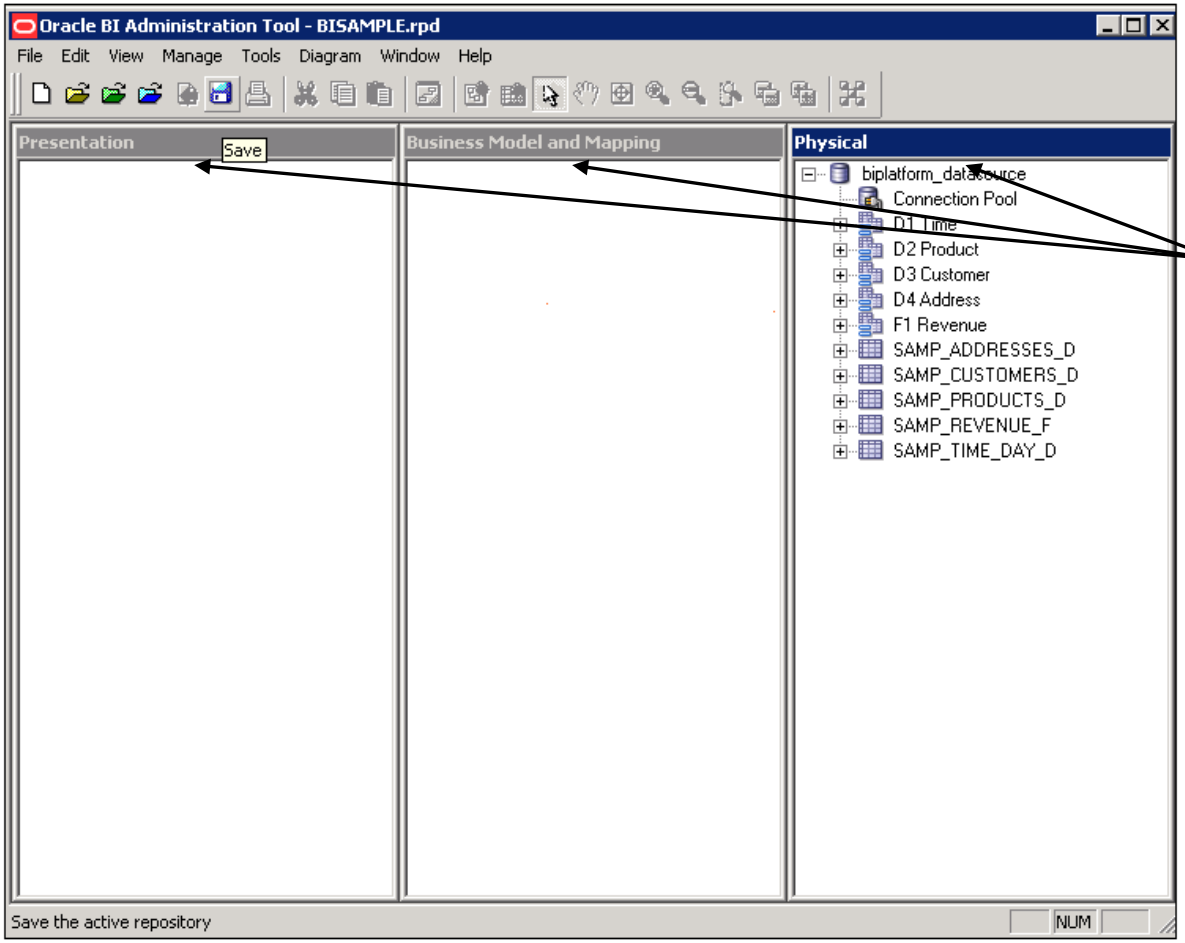


Oracle BI Repository details

- The repository relies on three layers:
  - **Physical layer**: this layer describes individual data sources i.e. OLTP databases, OLAP and other non-relational data sources. The way of accessing them and tables of interest within them are identified.
  - **Business model and mapping layer**: in this layer, the properties of logical tables, columns and dimensions are defined. This includes defining measures and the way of aggregating them, expressions based on raw columns, or the categories of join operations linking the tables.
  - **Presentation layer**: in this layer, presentation settings such as multiple subject areas grouping presentation tables can be defined.

1. To define a repository, we start from defining the content of its physical layer, to move on to business model and mapping layer and finish at presentation layer.
2. These operations are performed with Oracle BI Administration tool.

Oracle BI Repository in Oracle BI Administration tool



The repository layers are reflected in Oracle BI Admin tool interface.
In this case, physical layer has been defined. The remaining layers still have to be defined.

Recently Oracle introduced Semantic Modeler, a web-based tool for creating semantic models and publishing them as an RPD file for deployment.
https://blogs.oracle.com/analytics/post/the-semantic-modeler-in-oracle-analytics-cloud

Physical Layer

Physical layer: importing metadata

- Oracle BI can import metadata and accessing data sources of varied categories. Key categories include:
  - Relational data sources
  - Multidimensional data sources
  - XML data sources

1. Importantly, the role of Business Intelligence software is to provide standardised user interface for developing reports with variety of underlying data sources.
2. Oracle BI makes it possible to import metadata such as the structure of tables, columns and their data types, primary and foreign keys, but not only. In this way, development of reports is largely simplified.
3. Oracle BI is an example of such a platform providing access to relational, OLAP and other data sources.

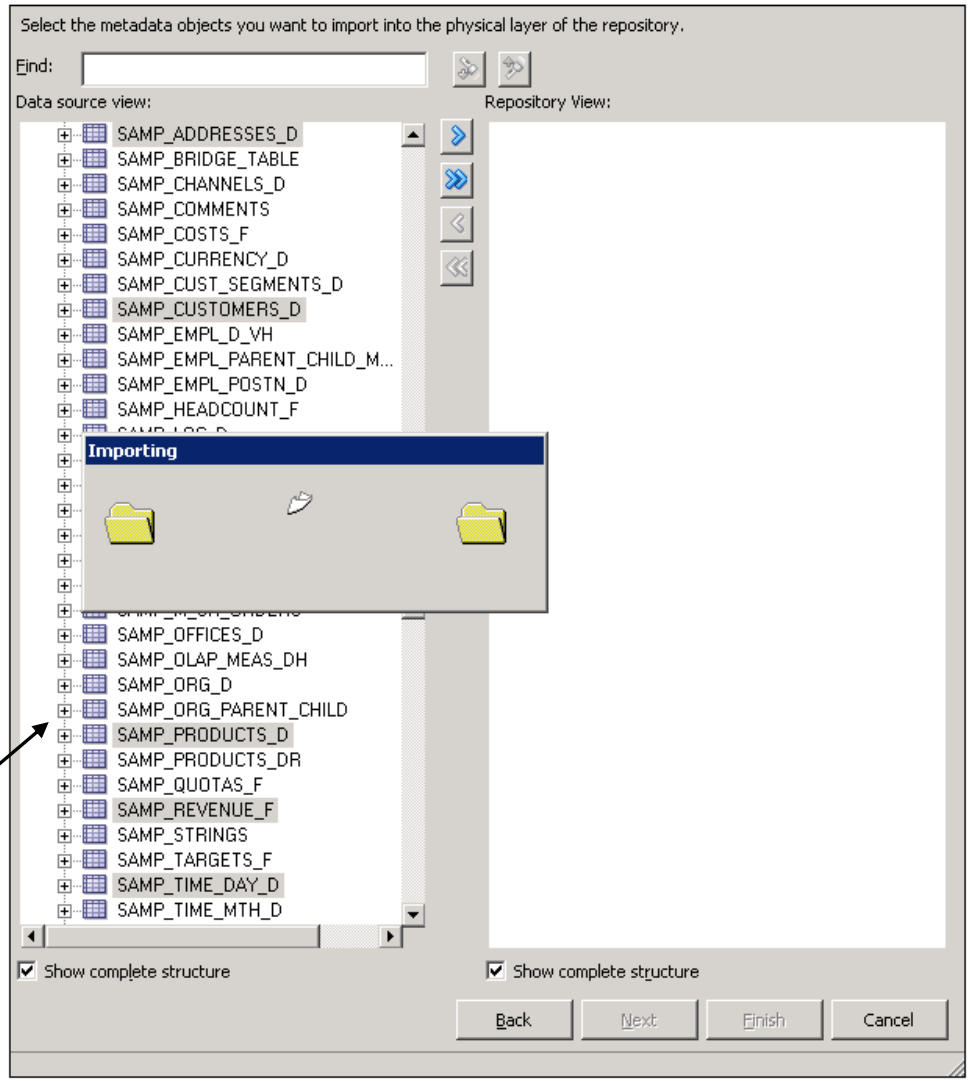
Physical layer: source platforms

- Some of the source platforms include:
  - Platforms available via ODBC (e.g. SQL Server)
  - Oracle Database
  - Essbase (Oracle OLAP solution)
  - Hyperion Financial Management
  - SAP BW (SAP Business Warehouse)
  - Teradata (one of key data warehousing solutions)

Physical layer: implementation

1. Physical layer constitutes the starting point for the development of a repository
2. It is typically populated with the metadata extracted from data sources such as Oracle or MS SQL Server databases

The definitions of these tables have been automatically extracted from an underlying data base



The content of physical layer: relational data sources

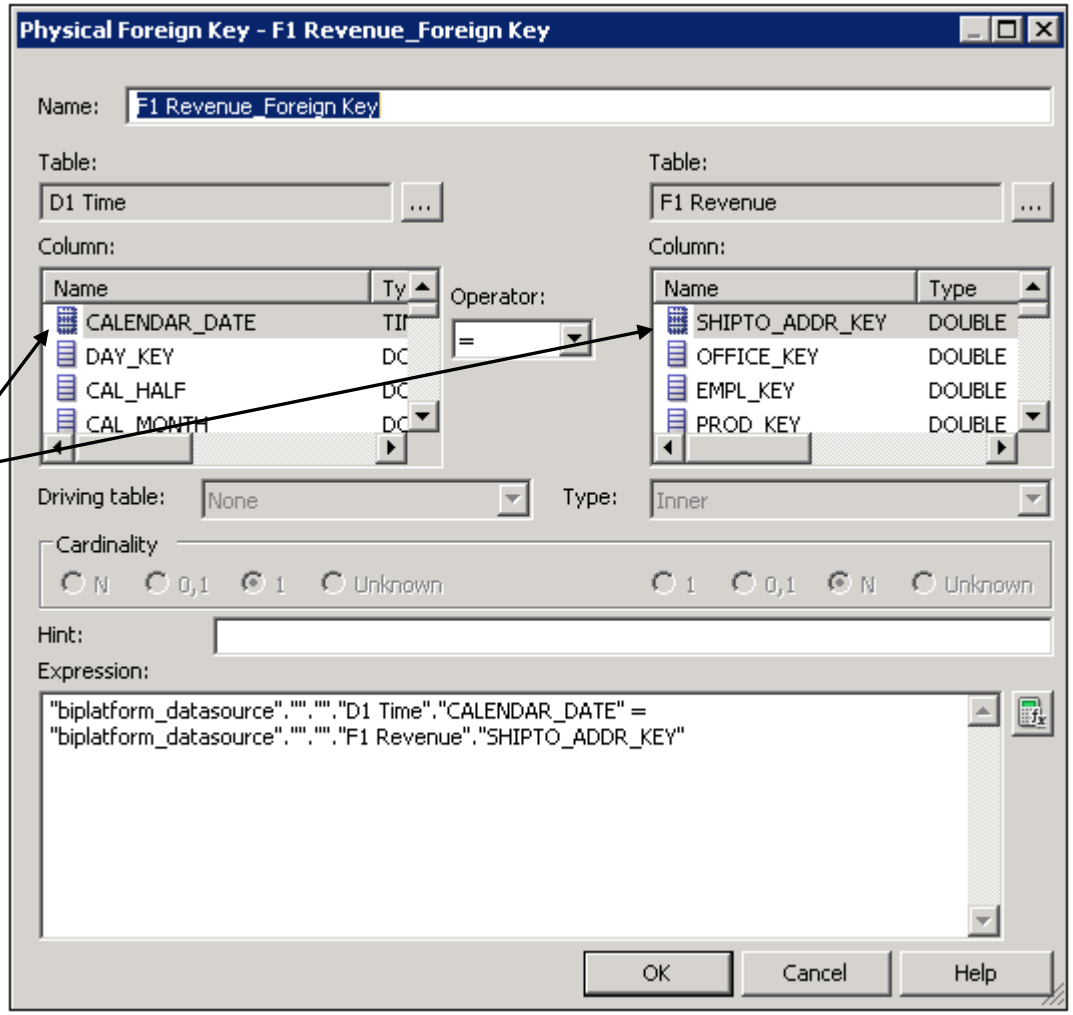
- For relational data sources, the physical layer contains a list of tables that were selected by repository designer to be used in Oracle BI.
- For every table, individual columns, primary and foreign keys can be extracted from a data source.
- In addition:
  - In DBMS, some of primary keys may be not defined. Hence, in physical layer primary keys possibly not defined in underlying data sources can be defined
  - In DBMS, some of foreign keys may be not defined. This can happen because of:
    - Design decisions of OLTP/OLAP system vendors
    - Performance reasons
  - Still, foreign keys possibly not defined in underlying data sources can be defined in the physical layer

The need for physical layer

- The settings in physical layer can be imported from metadata of underlying data stores. Even more importantly, they can be further refined and extended.
- Examples include:
  - Role-playing dimensions: in the case of dimension tables used in different roles such as Date dimension:
    - Multiple aliases (as many as the number of roles) can be set up.
    - For instance: DIM_Order_Date, DIM_Delivery_Date, and DIM_Payment_Date can be set up as aliases for DIM_Date table.
    - Foreign keys pointing to such aliases representing role-playing dimensions can be set. Typically setting a foreign key pointing to a view is not possible in a DBMS.
  - The ability to define foreign keys and complex joins between tables present in different data sources. Setting a foreign key to a table present in a database residing in another RDBMS may be impossible at a DBMS level
  - The ability to define aliases i.e. more descriptive names of tables to be used instead of physical table names present in data sources

Role-playing dimension example

A foreign key is set, which links Calendar_Date column with Shipto_Addr_Key



The content of physical layer: multidimensional data sources

- In the case of multidimensional data sources, the repository contains the following key object types:
  - Cube table i.e. the table reflecting physical cube columns, dimensions and other objects
  - Cube columns i.e. columns derived from measures or hierarchical dimensions
  - Cube dimensions i.e. dimensions reflecting the dimensions existing in data cubes present in data sources
  - Physical hierarchies i.e. hierarchies defined e.g. for date dimension or location dimension
- These objects are extracted from metadata contained in a data source

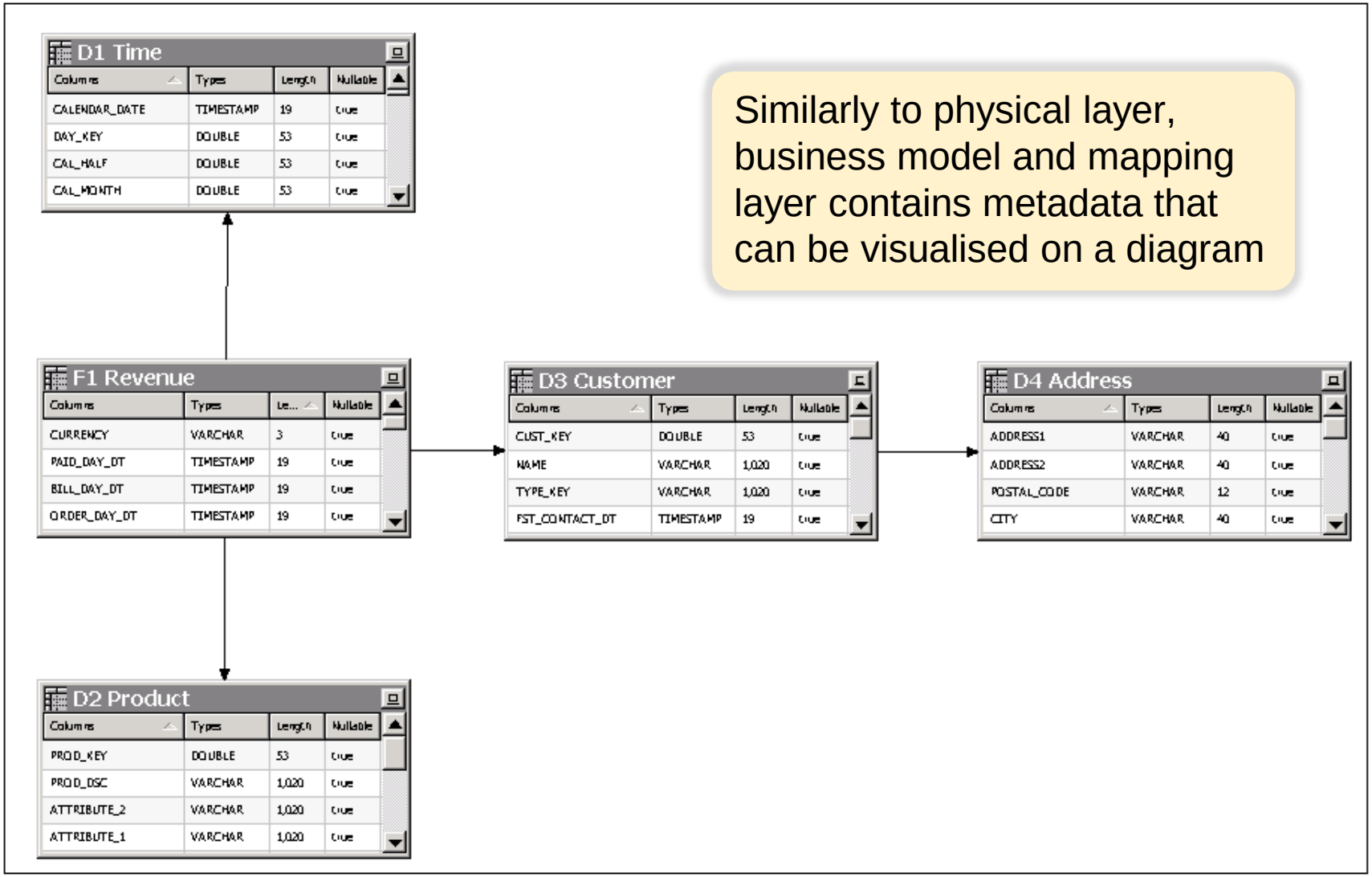
Business model and mapping layer

Possible content of Business Model and Mapping Layer

Options	Comments
Multiple business models, each of them linked to one data source in the physical layer	1. This option takes place when data sources are not integrated 2. This is not a recommended setting
One business model integrating the data from multiple data sources in physical layer	1. This is the recommended option. The consequence is that Oracle BI users can access what looks like a single model. In fact, such a model spans multiple subject areas 2. The integration of individual areas and data sources is attained through conformed dimensions
A mixture of both approaches with some business models relying on a single data source and some of them spanning multiple data sources	1. Can be seen as a step towards a single, centralised model

Conformed dimensions are crucial for integrated data model. This confirms their role in design process.

Business model visualised



The role of Business Model and Mapping layer

- In this layer:
  - Measures and the way of calculating them e.g. averaging a certain column can be defined
  - Expressions e.g. tax calculation can be defined
  - Hierarchies can be defined for individual dimensions
  - Logical tables integrating the data from multiple tables possibly coming from different data sources can be defined
  - Join categories can be clarified (LEFT JOIN, INNER JOIN etc.) to properly link individual tables

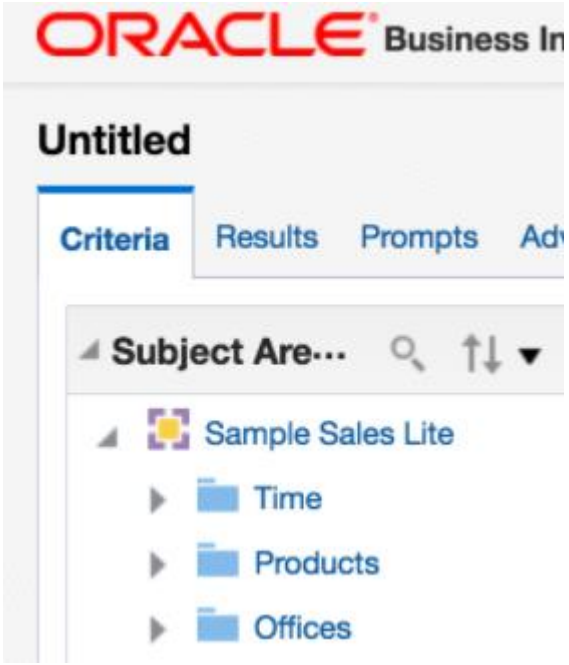
Presentation layer

The role of Presentation layer

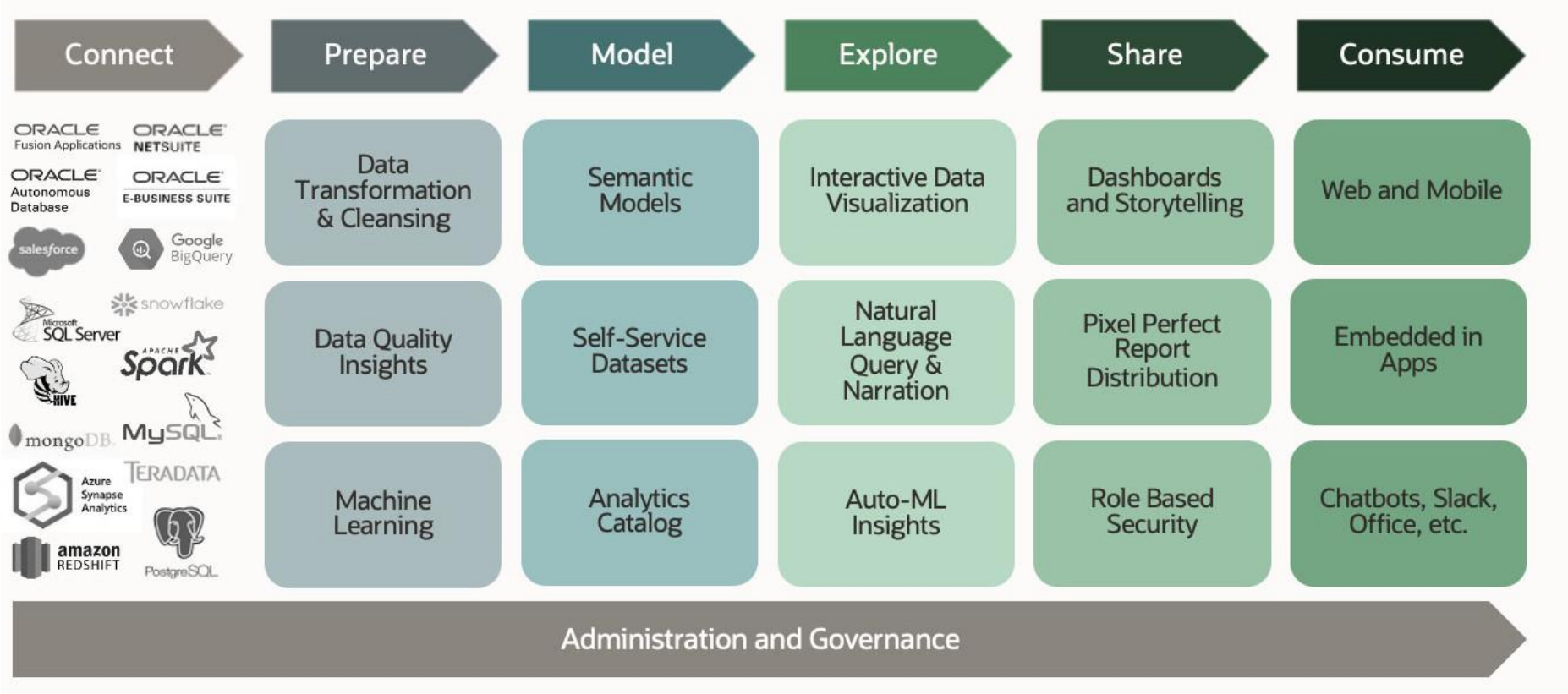
- Exposing one large business model to all business users would increase the complexity of using it
- Instead, multiple subject areas can be defined in the presentation layer. A subject area can group a fact table and dimension tables linked to it.
- Hence, a user can access the subject area he/she is interested in such as Orders, Procurements or Complaints even though all of them can rely on the same business model.
- Therefore, single dimension table is very likely to appear in many subject areas.

The role of Presentation layer

- The presentation layer is relatively the simplest one to populate with the data.
- Frequently, it can be easily populated with the settings based on business model and mapping layer content.
- It is the presentation layer that will be accessed in the remaining Oracle BI tools i.e. by end users to create their reports.



Oracle Analytics Cloud



Source: https://docs.oracle.com/en/cloud/paas/analytics-cloud/acsgs/what-is-oracle-analytics-cloud.html

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zadanie 10 pn. Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych, realizowane w ramach projektu ,,NERW 2 PW. Nauka - Edukacja - Rozwój - Współpraca'', współfinansowanego ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

Data visualization in Business Intelligence systems

Jakub Abelski, M.Sc.
J.Abelski@mini.pw.edu.pl

Hurtownie danych i systemy Business Intelligence
Wydział Matematyki i Nauk Informacyjnych
Politechnika Warszawska

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

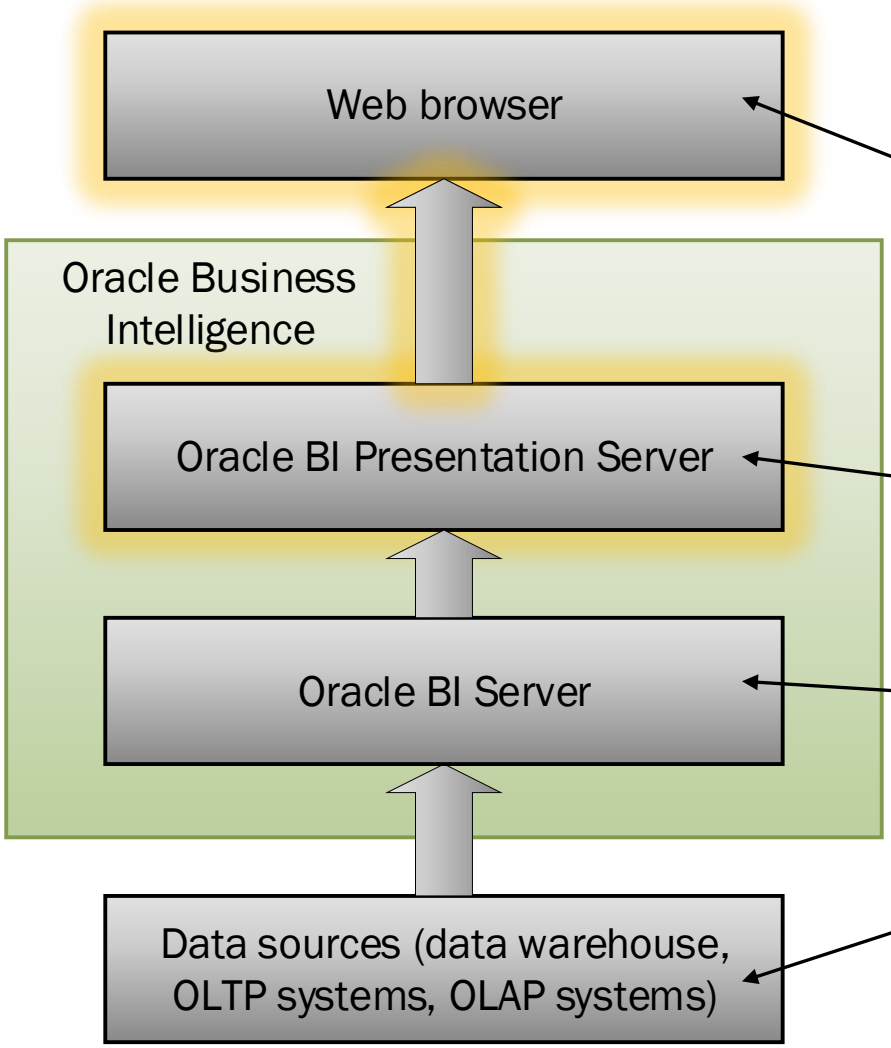
Zadanie 10 pn. Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych, realizowane w ramach projektu ,,NERW 2 PW. Nauka - Edukacja - Rozwój - Współpraca'', współfinansowanego ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

References

- **[Oracle2020]** *User's Guide for Oracle Business Intelligence Enterprise Edition*, Oracle, 2020 (available at: https://docs.oracle.com/middleware/bi12214/biee/BIEUG/BIEUG.pdf)
- **[Powell2018]** Powell B., *Mastering Microsoft Power BI: Expert techniques for effective data analytics and business intelligence*, First Edition, Packt Publishing, 2018
- **[SAS2020]** SAS Institute, *SAS® Visual Analytics for SAS® Viya*, First Edition, SAS Institute, 2020

BI user perspective – example based on Oracle Business Intelligence



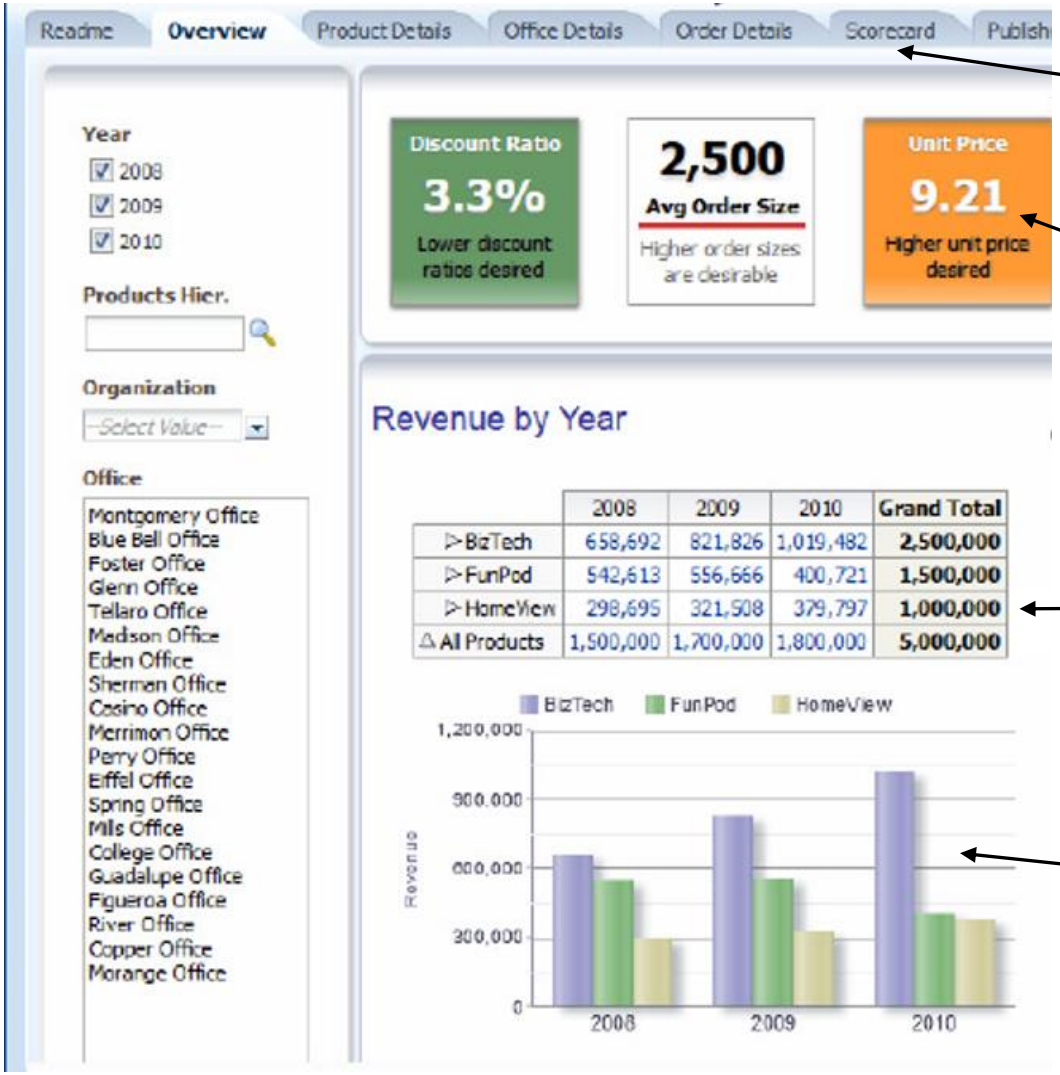
Used by end users including IT users and business users to access dashboards, reports and other analytical artefacts. **This relies on variety of view types.**

This server connects to the underlying layer to handle dashboards, analyses and reports

This server provides simultaneous access to various data sources.

Data sources of different categories can be used by BI solution

Sample view types



Dashboard with a number of tabs

Performance tiles

Interactive tables with drill-down capabilities

Interactive figures with drill-down capabilities

Visualisation view types (1/4)

View type	Description	Comments
Table	A classic tabular view showing the data in the form of a table. Dimensions and measures are columns. Facts or aggregated facts are rows	• A default tabular form, especially useful for relatively small quantities of data not described with hierarchical dimensions
Pivot View	Extends interactive capabilities by adding the possibility to display multiple levels of both row and column headings. This relies on hierarchical dimensions	• This form of tables is particularly useful for displaying large volumes of data, drilling down along hierarchical dimensions and investigating trends
Performance Tile / KPI	Displays a single measure value in the way focusing attention	• This form is especially useful for displaying major KPI values • The colour of the text and background can be set to match the importance and interpretation (positive or negative) of the value displayed

Visualisation view types (2/4)

View type	Description	Comments
Treemap	A two-dimensional visualization for hierarchical structures with multiple levels. Rectangular space is divided into a number of rectangular tiles.	• Relies on the values of two measures – one for defining the size of individual tiles, the other for defining the colour of a tile
Heat Matrix	Displays a two-dimensional depiction of data in which values are represented by a gradient of colours	• Well-suited for analysing large amounts of data and identifying outliers • Data structure similar to pivot tables (formed by the grouping and intersection of rows and columns)
Graph / Chart	Graphs of different categories are available (bar, line, scatter, bubble, radar…)	• Typically used to visualise fact data • Measures are the values to show, dimensions provide basis for filtering and labelling the data series.

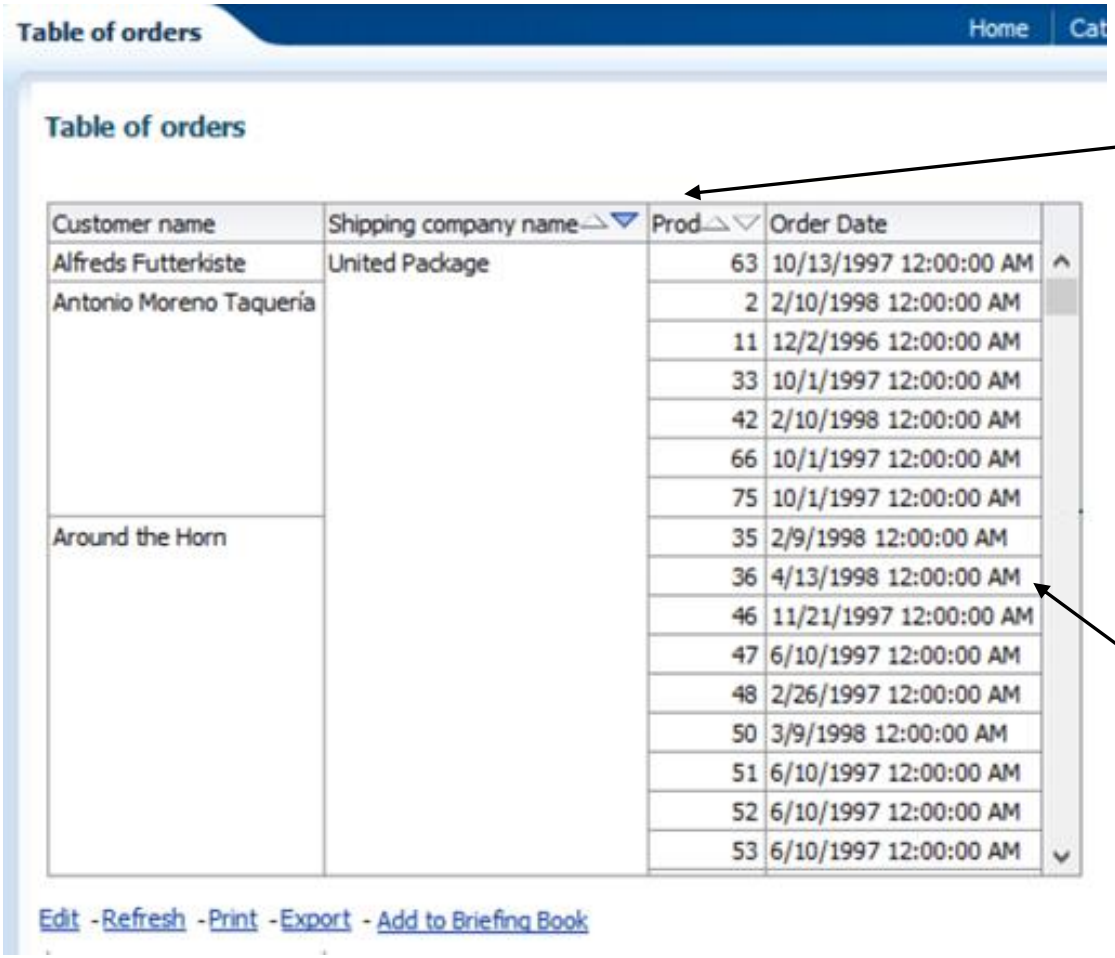
Visualisation view types (3/4)

View type	Description	Comments
Funnel	Used to visualise iterative processes such that at each stage the value of a measure gets lower	• Example: produced products, products delivered, products accepted by the client, products paid for
Gauge	A graph showing the status of a single measure e.g. the overall volume of sales in a certain country. Hence, a single aggregate of a single measure is visualised and mapped to colour scale	• Typically, green area is used for acceptable/desired values, yellow for warning zone, red for unacceptable values such as too long delivery times or too low volume of sales
Map view	Used to display results overlain on a map	• Depending on the data, the results can be overlain on top of a map as formats such as images, colour fill areas, bar and pie graphs, and variably sized markers

Visualisation view types (4/4)

View type	Description	Comments
Trellis	Can be defined as a grid (a table) containing multiple graphs inside. Exists in many forms: • **Simple:** includes a single graph type. Inner graphs have a synchronised scale and enable easy graph comparison • **Advanced:** contains a different inner graph type for every measure selected to be visualised. Moreover, each measure column is to large extent independent, for instance in terms of axis scaling	• Represents composite view type • Useful for displaying the same type of a figure for e.g. various countries to enable easy investigation of trends • Rely on trellis-dedicated artefacts such as micrographs

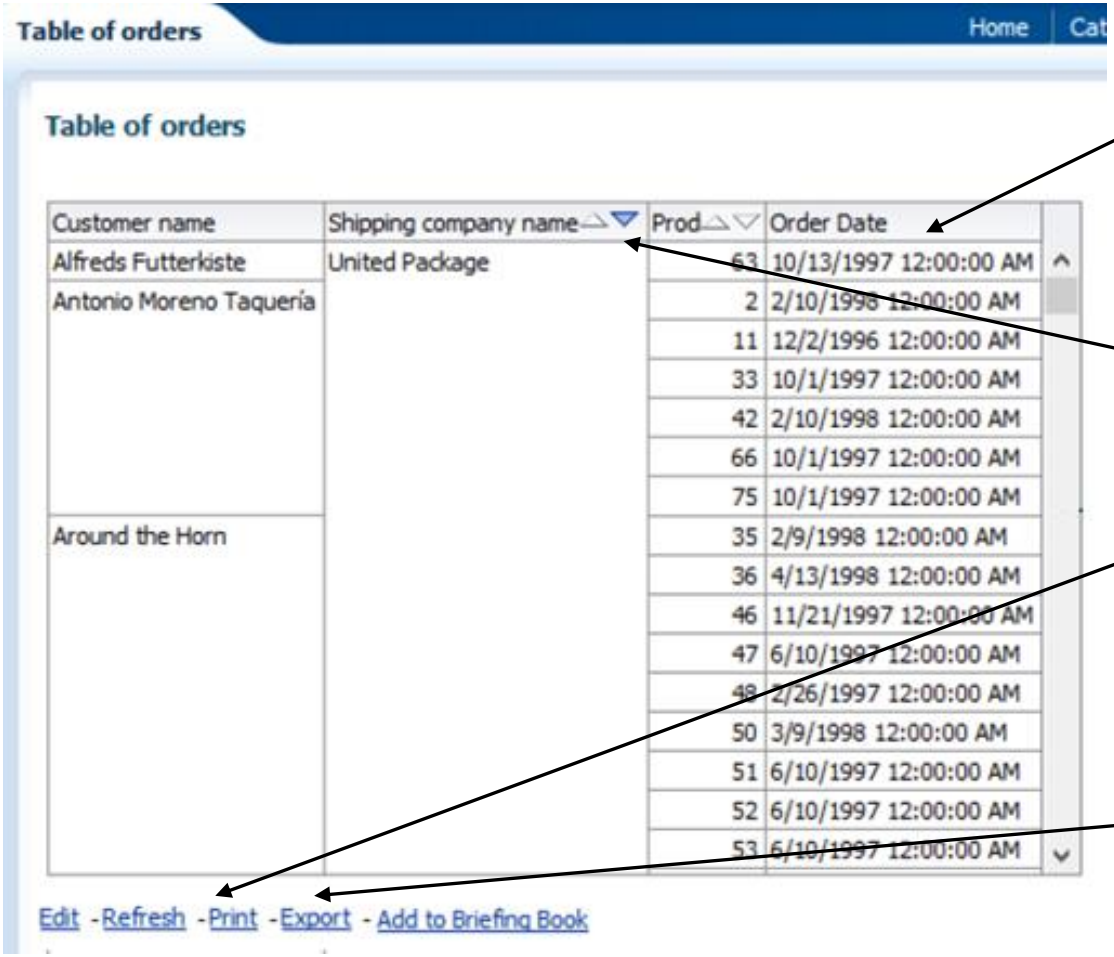
Generic table view



Multiple columns, typically coming from a fact table and dimension tables linked to it can be selected to be shown in the table

The queries to underlying database systems are automatically generated. Please note that they extract data from possibly many tables linked via JOIN statements. This relies on foreign keys and join categories defined in a repository.

Generic table view – modification and export



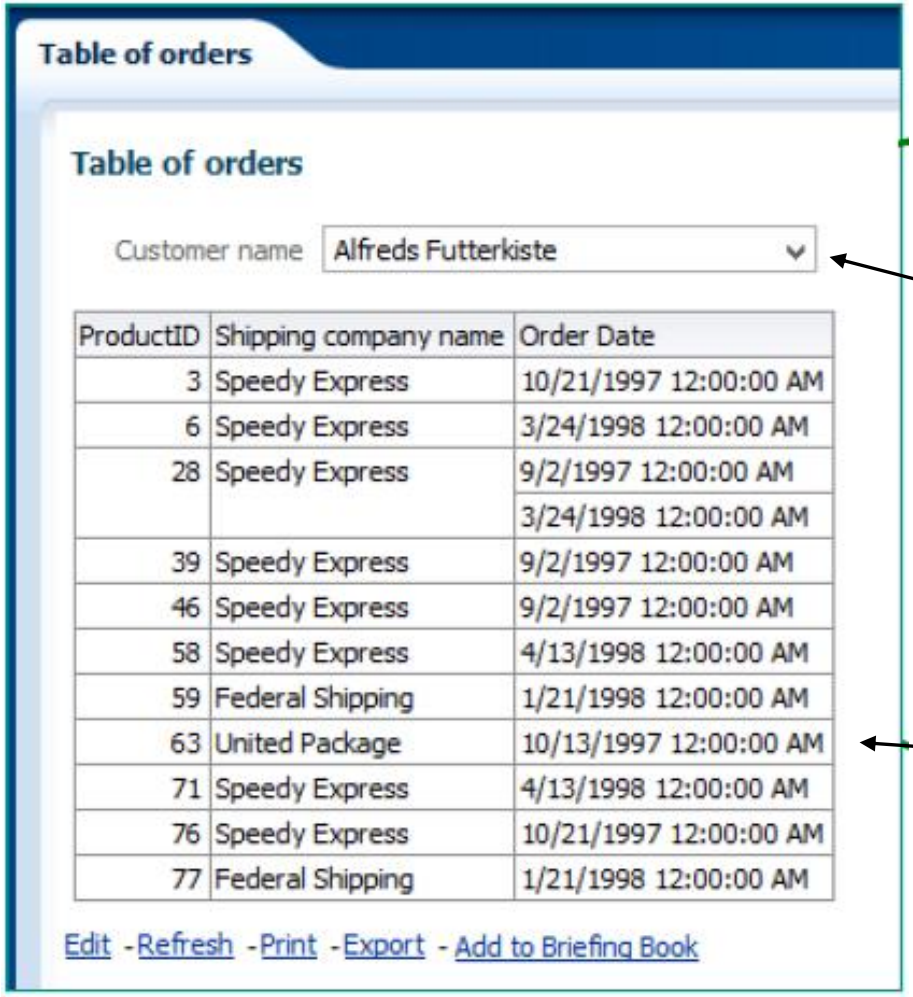
Columns can be dragged to another location in the table and temporarily excluded from the table.

Columns can be sorted

PDF and HTML version can be generated

The data in the table can be exported to Ms Excel file, XML, CSV, Ms PowerPoint etc.

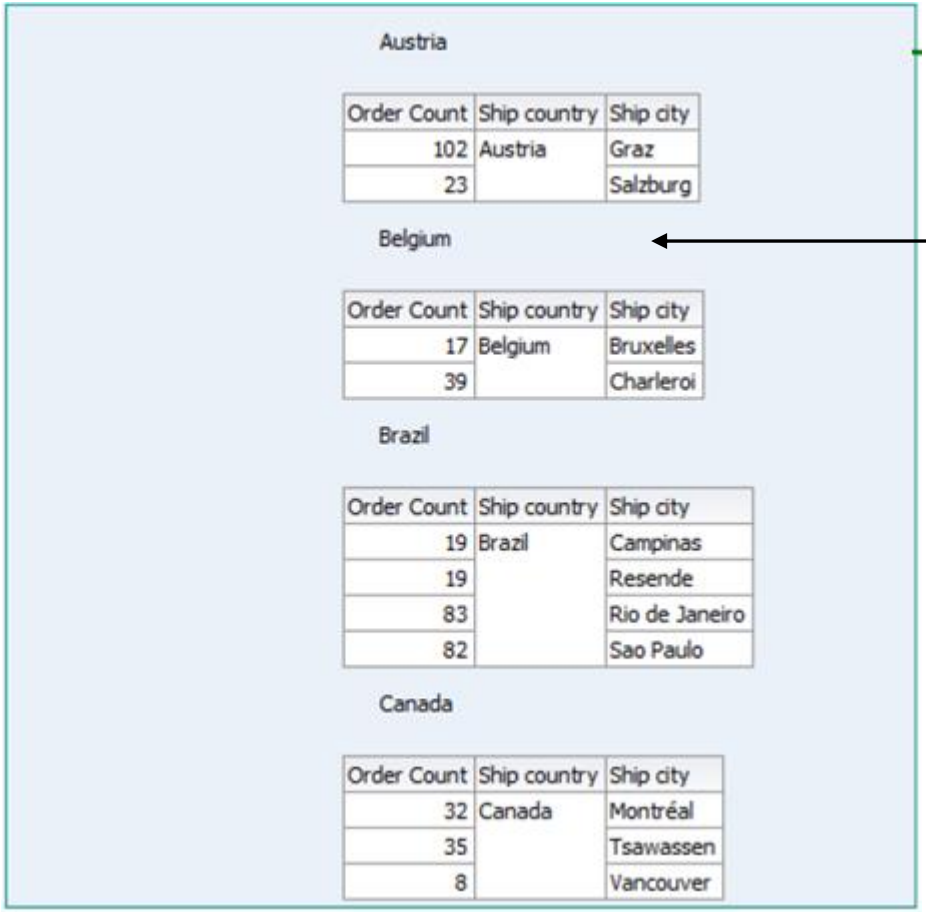
Generic table view – prompts



One or more columns can be moved to prompts, which means that they will be rendered as drop-down lists.
In this way, the value of this column selected by a user will be used to filter the data in the table.

Once prompt value is defined (here: customer name = 'Alfreds…'), only orders of this client are shown.

Generic table view – grouping



Austria

Order Count	Ship country	Ship city
102	Austria	Graz
23		Salzburg

Belgium

Order Count	Ship country	Ship city
17	Belgium	Bruxelles
39		Charleroi

Brazil

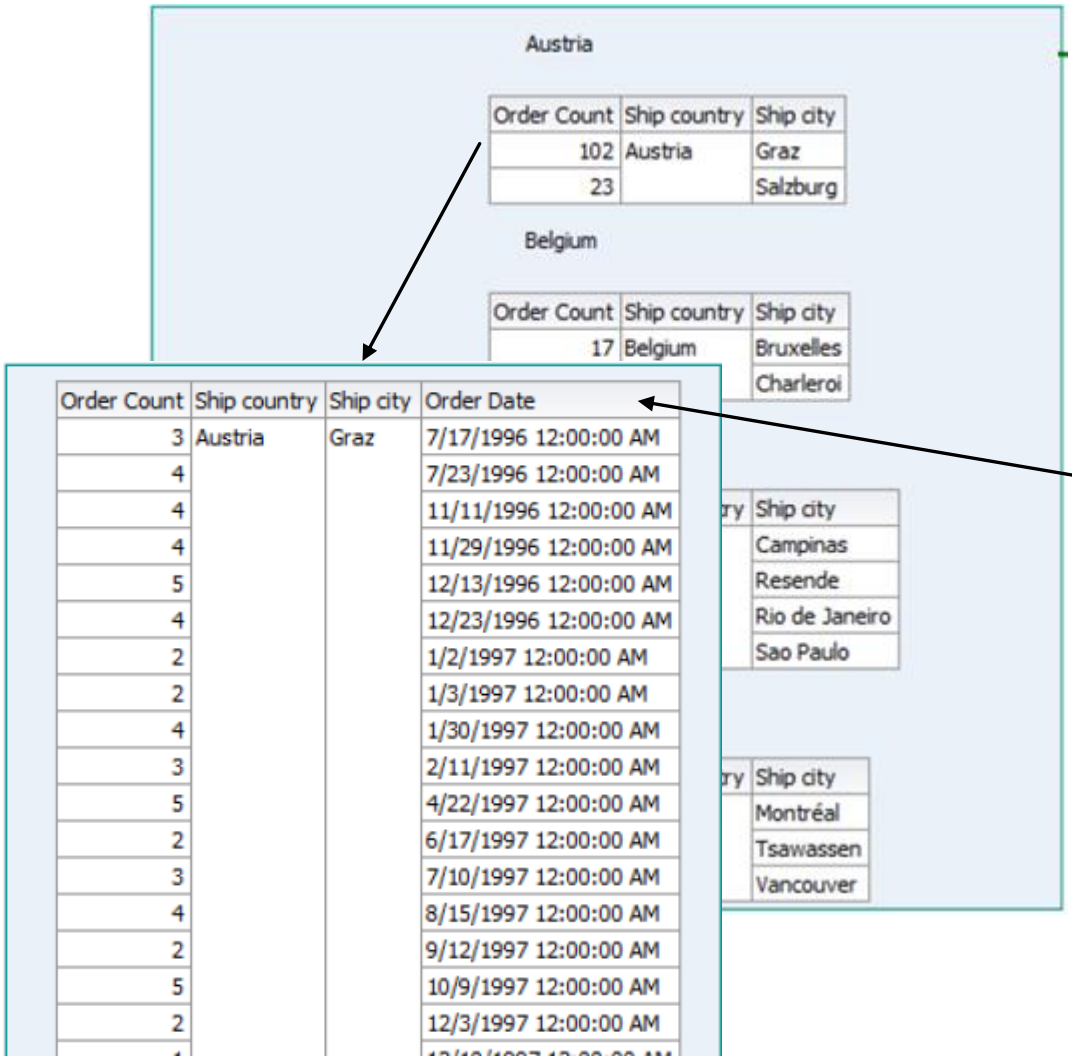
Order Count	Ship country	Ship city
19	Brazil	Campinas
19		Resende
83		Rio de Janeiro
82		Sao Paulo

Canada

Order Count	Ship country	Ship city
32	Canada	Montréal
35		Tsawassen
8		Vancouver

Column values can provide basis for sections. In this case, for every country a customer comes from, the number of order facts, ship country and ship cities are shown

Generic table view – aggregation



The data is automatically aggregated based on unique values of dimensions selected. In the analysed case, once Order Date is added to the list of columns, the data gets grouped based on ship country, ship city and order date. This corresponds to automatic GROUP BY clause generation.

Key Performance Indicators

- Key Performance Indicators, frequently abbreviated as KPIs are indicators of the performance of an organisation.
- These can be:
  - The volume of sales
  - The proportion of sales plan that has been fulfilled
  - Customer satisfaction level
  - Churn level
  - The proportion of orders delivered in time
  - The average time to deliver an order
- From BI perspective, all KPIs are aggregate measures calculated on the top of fact data and possibly dimension data

Gauge

The number, colour and range of individual sections can be defined.

In this case, sales efficiency per country basis is shown. This form of visualisation is ideal for showing aggregate data (e.g. KPI values).

Scorecard

Delay Performance Watchlist

Objects ▾ View ▾ Summary: ⚠ Warning (1) ✗ Critical (2) ✓ OK (1)

Label	Status	Trend	Actual	Target	% Variance	% Change
Delay as % of Flight Time	⚠		8.93	9.00	-0.76%	
Short Rtes Delay %	✗		15.37	9.00	70.83%	
Med Rtes Delay %	✗		9.07	9.00	0.75%	

Delay Growth After Depart

Objects ▾ View ▾ Summary: ✗ Critical (1) ✓ OK (3)

Label	Status	Trend	Actual	Target	% Variance	% Change
Short Rtes Delay Growth	✗		5.99	5.00	19.73%	
Med Rtes Delay Growth	✓		3.08	5.00	-38.37%	
Delay Growth after Take Off	✓		3.30	5.00	-33.94%	

The scorecard is a visual that enables putting many different KPIs together to give a valuable insight on the business performance.

Performance Matrix

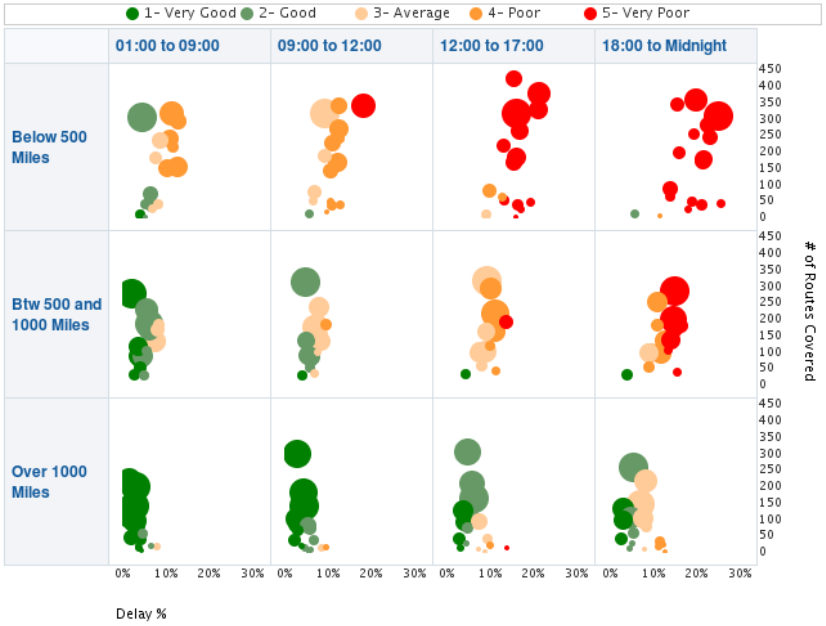
Airlines Delay Performance Matrix
By Distance Group by Departure Time

Delay Perf % (0=Ontime, >0=Late)

	01:00 to 09:00	09:00 to 12:00	12:00 to 17:00	18:00 to Midnight	Grand Total
Below 500 Miles	9.0%	11.7%	16.9%	21.1%	**15.4%**
Btw 500 and 1000 Miles	5.1%	6.7%	10.1%	13.1%	**9.1%**
Over 1000 Miles	2.8%	4.0%	5.6%	6.2%	**4.8%**
Grand Total	**5.1%**	**6.8%**	**10.2%**	**12.4%**	**9.0%**

# of Flights

	01:00 to 09:00	09:00 to 12:00	12:00 to 17:00	18:00 to Midnight	Grand Total
Below 500 Miles	560,202	511,684	858,093	742,362	**2,672,341**
Btw 500 and 1000 Miles	469,324	408,820	687,098	555,666	**2,120,908**
Over 1000 Miles	317,285	277,646	406,453	359,250	**1,360,634**
Grand Total	**1,346,811**	**1,198,150**	**1,951,644**	**1,657,278**	**6,153,883**



BI tools include various view types suitable for the visualisation of aggregated data.

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zadanie 10 pn. Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych, realizowane w ramach projektu ,,NERW 2 PW. Nauka - Edukacja - Rozwój - Współpraca'', współfinansowanego ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

DW/OLAP/BI deployment considerations

Jakub Abelski, M.Sc.
J.Abelski@mini.pw.edu.pl

Hurtownie danych i systemy Business Intelligence
Wydział Matematyki i Nauk Informacyjnych
Politechnika Warszawska

Fundusze Europejskie
Wiedza Edukacja Rozwój

Rzeczpospolita Polska

Politechnika Warszawska

Unia Europejska
Europejski Fundusz Społeczny

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zadanie 10 pn. Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych, realizowane w ramach projektu ,,NERW 2 PW. Nauka - Edukacja - Rozwój - Współpraca'', współfinansowanego ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

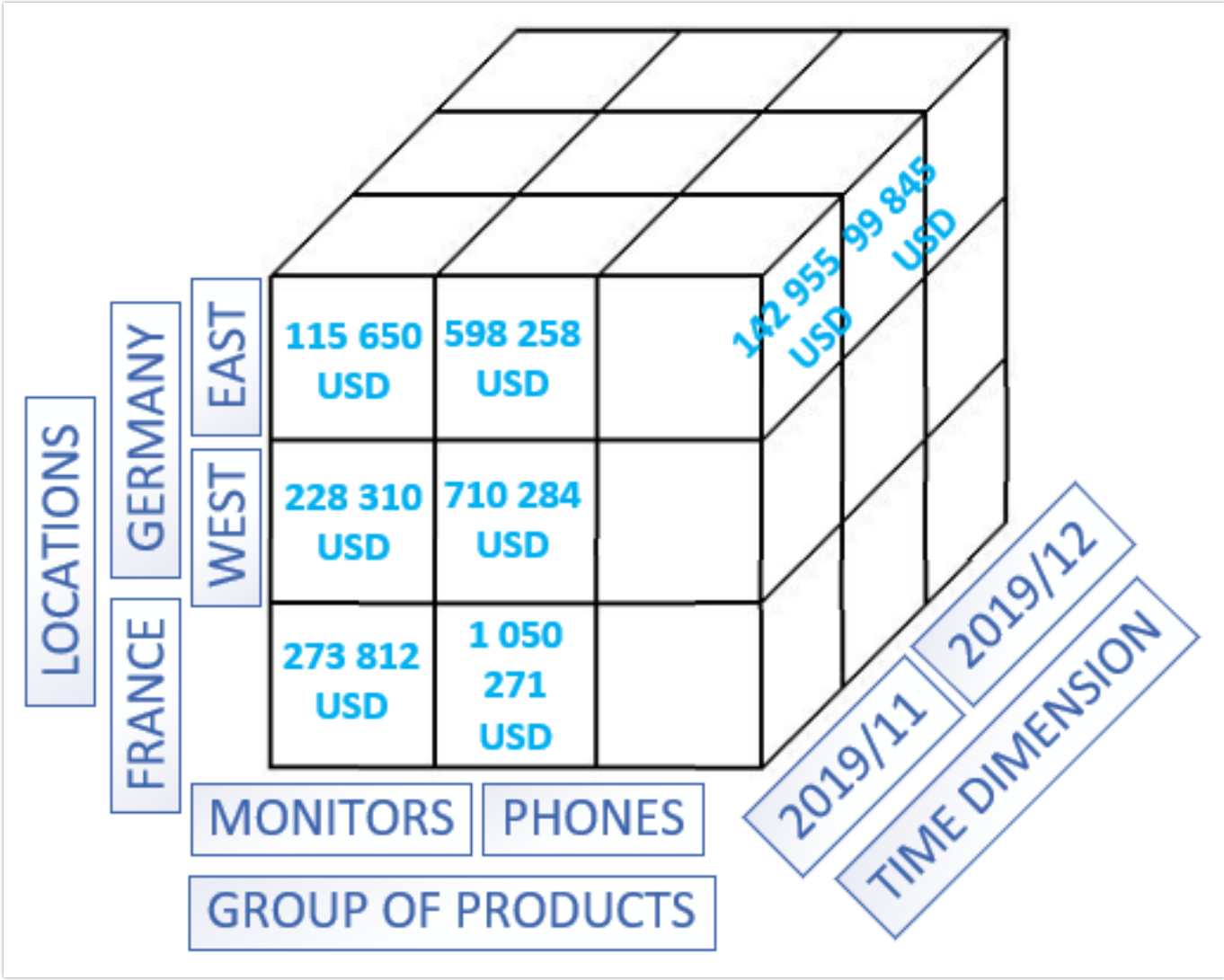
Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

References

- [**Howson2014**], C. Howson, *Successful Business Intelligence, Second Edition: Unlock the Value of BI & Big Data*, McGrow Hill Education, 2013
- [**Kimball2013**] Kimball, R., Ross, M., *The Data Warehouse Toolkit. The Definitive Guide to Dimensional Modelling*, Wiley, 3rd Ed., 2013
- [**Rittman2013**] Mark Rittman, *Oracle Business Intelligence 11g Developers Guide*, Oracle Press, 2013
- [**Root2012**] Root R., Mason, C., *Pro SQL Server 2012 BI Solutions*, Apress, 2012
- [**Ward2017**] Adrian Ward, Christian Screen, Haroun Khan, *Oracle Business Intelligence Enterprise Edition 12c*, Second Edition, Packt Publishing, 2017

OLAP data cube revisited

A cube is a primary data structure defined in data warehouses and managed by OLAP systems to ensure efficient data processing.



OLAP cubes vs. star model

	Star model	OLAP cube
Implemented in	RDBMS	Multidimensional database environments
Data storage	Just a collection of tables in RDBMS. In particular, RDBMS is unaware of the definitions of hierarchical dimensions or measures and the way they will be aggregated. Aggregate values are discouraged.	Data is stored using formats and techniques designed for dimensional models. For instance, pre-calculated summary tables can be created and managed by OLAP engine. Also indexing strategies can better match the needs of dimensional model. OLAP engine fully knows measures and dimension structures used by the cube.
Analytical functions	Star model relies on SQL capabilities and the capabilities of BI layer to perform calculations.	OLAP cubes frequently provide more analytical capabilities than general purpose RDBMS tables. In addition, BI capabilities can be also used.

A frequent practice is to load data into a star schema and build OLAP cubes on the top of these data. Hence, Ralph Kimball and Margy Ross in [Kimball2013] discuss most dimensional modelling techniques on the example of star schema.

Terminology and implementation issues

- What is referred to by Ralph Kimball and Margy Ross as star model, is also known as reporting tables or simply data warehouse tables [Root2012]
- Similarly, reporting database can be the term used by some organisations instead of a data warehouse [Howson2014]
- A decision can be made to use:
  - The same RDBMS for data warehouse and its tables as for operational database(s) e.g. Oracle
  - Or a relational system such as Teradata or Netezza designed for data warehousing/BI needs

The pros and cons of star model and OLAP cubes

	Star model / RDBMS	OLAP cube / multidim. DBMS
Configuration	Less effort required	More effort required
Performance	Depends on data model structure and SQL queries optimization (improved by columnar store)	Optimized for business analysis in scope of the modelled data cubes
Migration	Easier to migrate a star model from one RDBMS to another	More complex to migrate cubes from one OLAP vendor to another
Analysis capabilities	The simplicity, but also limitations of SQL	Significantly extended by analytical languages such as MDX
Support for fact tables	All fact types supported	Potentially difficult for fact tables accepting data update
Support for hierarchies, calc. maasures, KPIs	Requires SQL implementation, lack of native mechanisms define a logical object representation	Build-in functionalities to work with hierarchies and complex calculation structures
Security	Possibility to configure row level security (RLS) and object level security (OLS)	Advanced settings such as preventing access to detailed data are in general easier in OLAP cubes

Case study
OLAP cubes in Microsoft SQL Server Analysis Services (SSAS)

As stated before, the capabilities on multidimensional database servers vary to a large extent. This is one of the reasons making migration of OLAP cubes more difficult than the migration of a star model deployed in RDBMS.

Microsoft Analysis Services offering

- Microsoft offers several variants of analytical platforms used in decision support and business analytics:
  - **SQL Server Analysis Services**: installed as an on-premises server instance providing enterprise-grade semantic data models for business reports and client applications. It supports tabular and multidimensional models, data mining, and Power Pivot.
  - **Azure Analysis Services**: fully managed platform as a service (PaaS) providing enterprise-grade data models in the cloud. It supports tabular model and provides ad hoc data analysis using tools like Power BI and Excel.
  - **Power BI Premium**: hosted service (SaaS) providing a superset of the Azure Analysis Services capabilities It allows to host datasets and share Power BI reports and dashboards with anyone in the organization without requiring recipients to be individually licensed. It is a premium offering for enterprise and self-service BI licensed by capacity.

The differences between SQL Server database and Analysis Server database

	SQL Server database	SSAS database
Category	RDBMS	Multidimensional database
Contents per Server	One SQL Server can host many databases	One Analysis Server can host many databases
Contents per database	Many tables	Many cubes and dimensions
Data storage	Many tables co-exist in the same database file (.mdf file). Changes to the tables are logged in .ldf file	Each database is a hard drive folder. Files and subfolders are physical representation of the cubes and dimensions
Transactional processing	Supported - transaction logging is a vital part of the system	Not supported - transaction logging is not available
Updates to the data	Supported - key aspect of running database	Possible - but most data is copied and once copied never modified

Creating a cube in SSAS

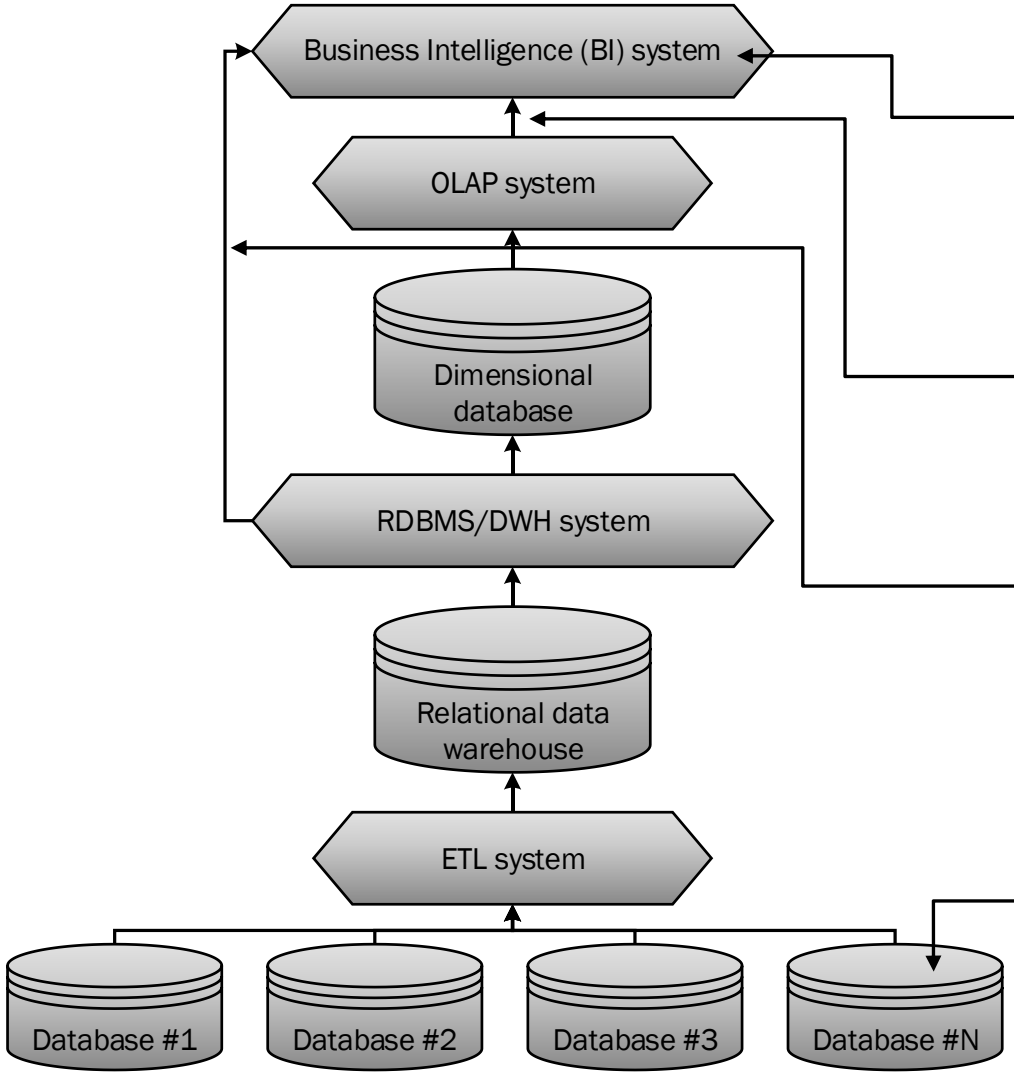
- Can be done:
  - Graphically, within an SSAS project created in MS Visual Studio
  - Programmatically using MDX or XMLA
- Technical steps required to create a data cube are performed during the labs
- It is important to note that the settings made in SSAS project in Visual Studio have to be deployed to SSAS server. In particular, the XML files created by Visual Studio are converted then to XMLA format.

MDX and DAX

- MDX:
  - Multidimensional Expressions used to query OLAP databases and create data cubes
  - Initiated by Microsoft, the ideas and solutions were adopted by many other vendors
  - No independent standard of the language
  - Reference documentation available at: https://learn.microsoft.com/en-us/sql/mdx/multidimensional-expressions-mdx-reference?view=sql-server-ver16
- DAX:
  - Abbreviation for Data Analysis Expressions
  - A library of functions and operators that can be combined to build formulas and expressions for tabular data models in Power BI, Analysis Services, and Power Pivot in Excel
  - DAX formulas are used in measures, calculated columns, calculated tables, and row-level security
  - Reference documentation available at: https://learn.microsoft.com/en-us/dax/

DWH/OLAP/BI deployment
Data Quality and Performance Testing

Possible complex DWH/OLAP/BI layout



BI tools are expected to be used primarily by business staff to navigate through the data warehouse/OLAP cubes and/or analyse the data.

Data cubes provide access to some of the data, but not necessarily all of the DWH data

Some data contained in DWH tables can be directly accessed from BI tools.

This is where source data is placed in by operational teams such as salespersons, accounting staff and everyone using OLTP system to support his/her everyday operations. The quality of the data may be different.

Data cleansing

- Data cleansing is a processes to validate and correct data by resolving corrupt, inaccurate, or irrelevant information to boosts the consistency, reliability, and value of data.
- In practice, data cleansing means:
  - Matching similar, yet different strings of values (e.g. brand names written with errors, city names not matching true city names, various forms of writing the same name: Avenue of 11th November, Avenue of Eleventh November, Avenue of 11th Nov. etc.)
  - Dealing with missing column values such as missing region names
  - Dealing with duplicates and data passing the retention period
  - Integrating noisy data from various sources, detection and dealing with invalid data and outliers

Data cleansing is an important step to build the data harmonization strategy where multiple data sources are combined into an integrated, unambiguous "golden copy" record that can be used by down-stream systems. Master data management (MDM) is a good example of a system focused on harmonization of diverse master data maintained by an organization.

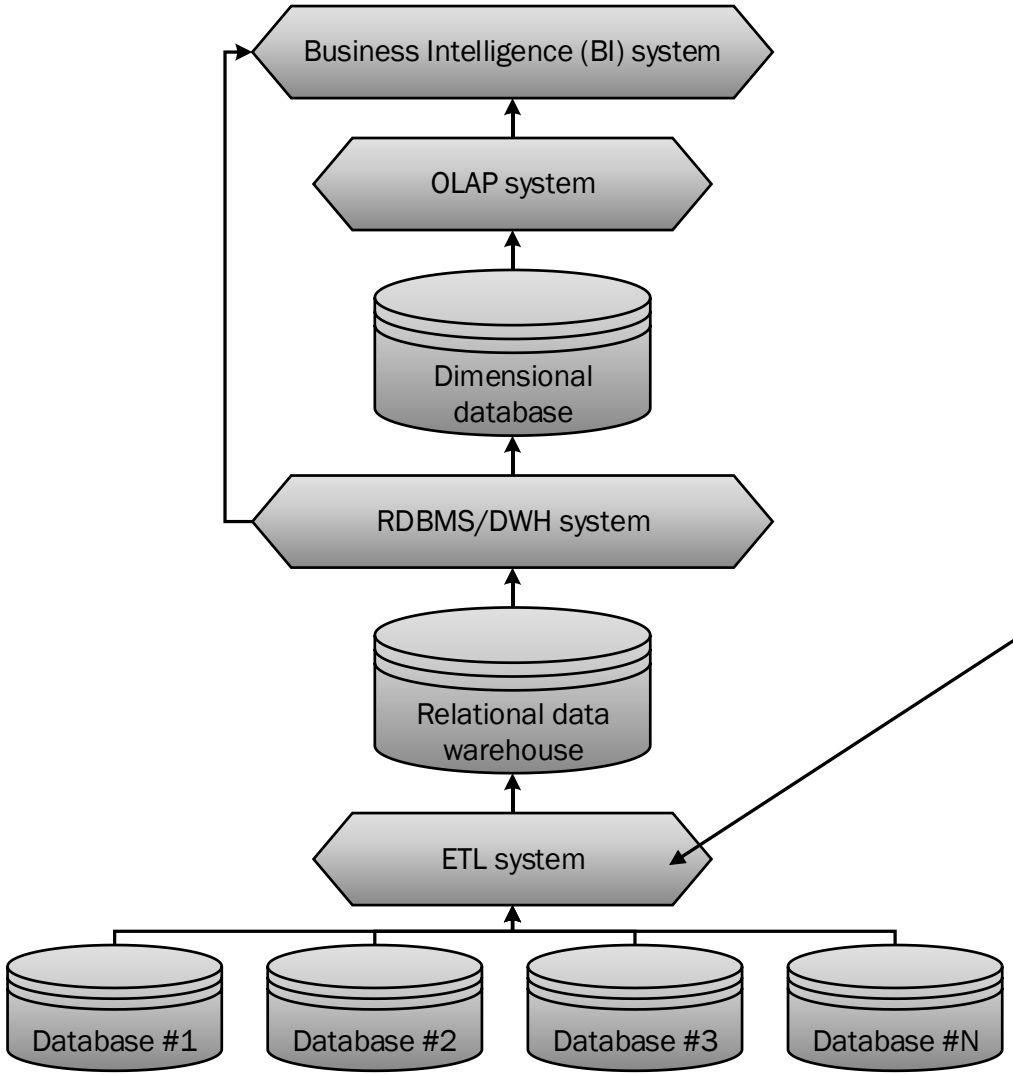
Sample data issues and their impact on analysis results

Problem	Possible impact
Orders made by inactive clients	A report showing only active clients will not show some of the orders. This will yield wrong aggregates by ignoring some sales facts.
Water consumption readings reported by non-existing meters	KPI showing the number of meters that were functioning properly will possibly show the number of operated meters higher than known to the company
The database of maintenance job orders claiming that a work unit worked on the replacement of two different pipes in two different villages at the same time of the day	The number of working hours spend by the team on job orders will get higher than the number of hours they worked within a month (and were paid for)
Orders submitted 1 year ago and still marked as in progress	The lag of not closed orders will get larger and larger. The true reason might be that the orders were cancelled or finished, but the status was not properly placed in operational database

The risk of using low quality data in DW/OLAP/BI

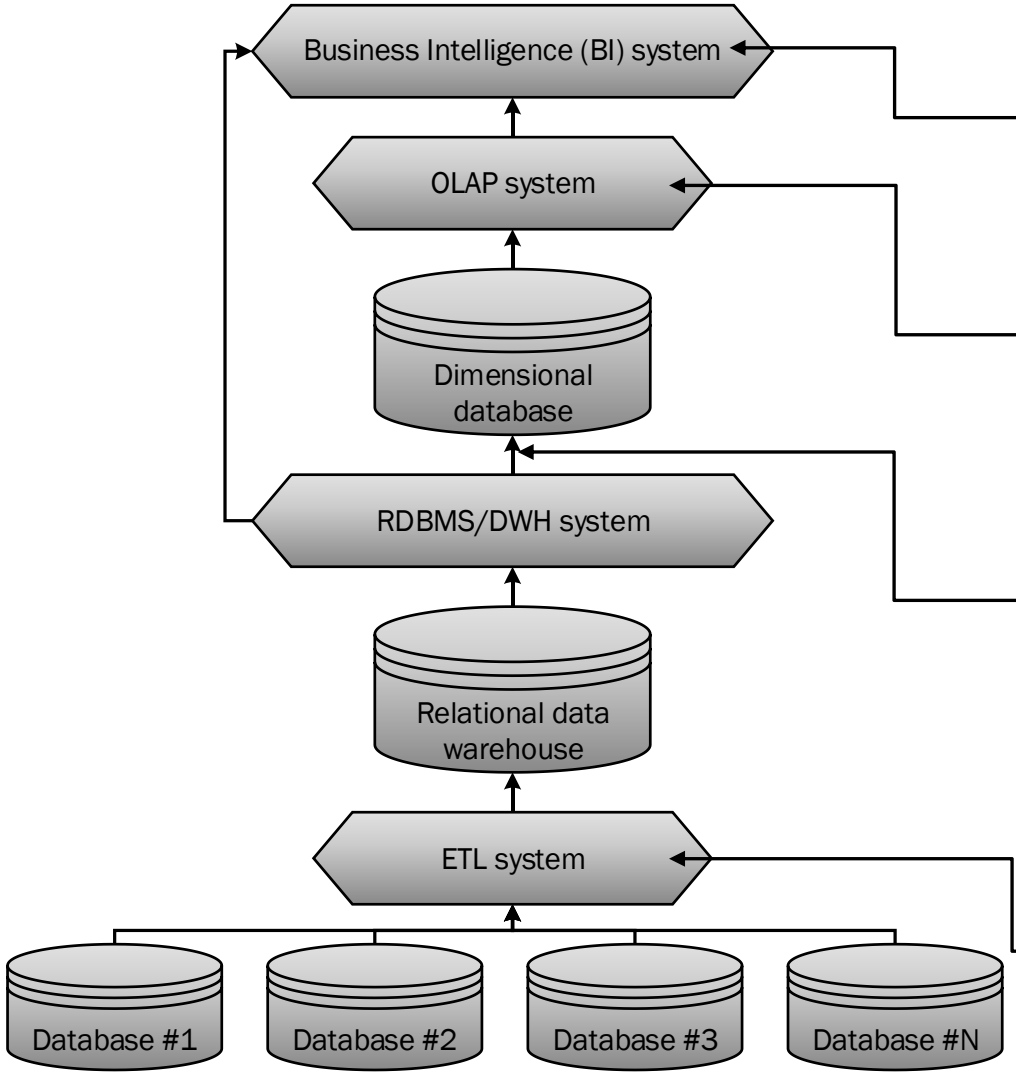
- Sometimes, **minor quality problems may have a major impact on KPI values**. For instance, if the processing of all service requests is supposed to be started within 4 hours since submission, even a single wrong request record may cause Service Level Agreement (SLA) violation.
- When the data served via BI system is of insufficient quality:
  - Wrong decisions can be made
  - The users are likely to:
    - Ignore or manually augment results with what they believe better reflects reality
    - Avoid using these systems despite all the investment made
  - Data-driven decision making can be replaced with intuitive decisions
- Hence, data cleansing is vital for high quality data analysis. Ideally, the quality of data should be maintained under the umbrella of end-to-end data management. In this way, many problems can be resolved in operational systems rather than using ETL to populate DW tables (avoid garbage-in garbage-out).

Possible complex DW/OLAP/BI layout
Data cleansing



Data cleansing is ideally done at this stage. This is to avoid placing wrong, inconsistent data in a data warehouse.

Possible complex DW/OLAP/BI layout
Performance considerations



BI tools may offer capabilities such as query caching, setting the maximum number of rows returned to end users or the expiry time of user sessions.

Reporting performance, which depends on OLAP engine performance.

Processing performance i.e. the performance of retrieving data from a data warehouse and putting it into dimensions and cubes.
Involves mostly SELECT and INSERT, but possibly also updates and deletes.

Performance of data retrieval (SELECT statements) and insertion (INSERT statements). Depending on fact table categories, INSERT and UPDATE statements may be also needed.

Selected aspects of performance testing

- Apart from other forms of testing, DW/OLAP/BI systems require:
  - **Performance tests**, to check the response time of critical functions, and possibly other aspects such as scalability and/or stability of the system. These measures can be observed when a single user is active and/or when many users are active in the system being tested.
  - **Load tests**, to verify whether the system can meet performance objectives such as the objectives expressed in Service Level Agreement. In this case, the system is tested under normal and peak load conditions.

These are not the only aspects of performance testing. Some other aspects include stress tests i.e. tests performed to observe the behaviour of the system beyond peak load conditions. See https://learn.microsoft.com/en-us/azure/architecture/framework/scalability/performance-test for further details of Microsoft recommendations in this area.

The role of performance testing in DW/OLAP/BI systems

- Performance tests among others aim at:
  - Ensuring that ETL processes will finish within available time slots. Importantly, this should happen also during high load periods e.g. high volume of sales periods.
  - Ensuring that the response time of DW/OLAP/BI systems will be acceptable also when multiple end users run the analysis and queries based on these systems
  - The systems will remain stable during longer periods of normal load. Load testing may reveal some resource leak problems that are not detected in individual requests but are observed after some time only
- Performance testing can be done by:
  - Saving the query trace with tools such as Ms SQL Server Profiler
  - Investigating response time for individual queries
  - Replicating the queries to simulate the system behaviour under requested number of queries/users

Data Cubes
Physical storage options

Data cubes and physical storage

- The data of data cubes can be stored in variety of ways:
  - As aggregates to speed up retrieval of aggregated data
  - In relational database including direct use of operational data to eliminate latency caused by importing the data into multidimensional databases
  - In the way trying to combine both approaches

OLAP cubes in Ms SQL Server Analysis services

- In MS SQL Server Analysis Services, the data of the cube can be divided into partitions
- There is at least one partition per a data cube
- Partitions:
  - Provide a way of setting different physical storage options for different subsets of cube data
  - Can be placed on different disks to increase performance
  - Can be configured in varied way. For instance, only some partitions can be configured to store data aggregations
  - Can be identified based on table and query binding i.e. the content retrieved from different tables or queries.

A possible option is to have for instance: current year facts in one partition, previous year in another partition and all the data preceding previous year in the third partition.

Partitions – storage options

- Three options can be set in SSAS:
  - MOLAP i.e. multidimensional OLAP
  - ROLAP i.e. relational OLAP
  - HOLAP i.e. hybrid OLAP
- These options are set independently for every partition of a data cube
- Even though three options are possible, the data of the partition is physically present in one or two locations i.e.:
  - At least partly in multidimensional database
  - Optionally also in relational database

The key difference between the three options is whether and if so to what extent the data of the cubes resides in relational database. Metadata of the partition is always stored in a multidimensional database.

MOLAP, ROLAP, and HOLAP compared

	Idea	Benefits	Drawbacks
MOLAP	Aggregations of the partition data and copy of its source data are stored in SSAS folder. **Default option.**	• Highly optimised to maximise query performance • No need to access data source to answer queries	• Data in MOLAP cube is as recent as the most recent processing of the partition. Periodical cube processing needed to reflect new data. • Latency in making most recent data available in data cube.
ROLAP	No copy of source data stored in SSAS folder. Aggregations are stored in index views in relational database. **Rarely used**.	• No need to copy any data to SSAS folders. • Most recent data shown in a data cube	• Longer query processing time and query response. • Constraints regarding the data organisation necessary to enable the creation of indexed views.
HOLAP	A combination of MOLAP and ROLAP. Only aggregations stored in multidimensional structure. No copy of source data stored in SSAS folder. These data reside just in relational database.	• No need to copy the source data to SSAS folders. • Useful when aggregated data is accessed most/all of the time e.g. in the form of KPIs.	• Not suitable, when raw data is accessed frequently. In such cases, MOLAP should be preferred.

SSAS cube settings: updates

- MOLAP and HOLAP cubes can be updated with new source data based on:
  - Near-real time manner, which is based on proactive caching. In this approach, notification process informing of changes happening in source data can be set up. By default, within 10 seconds after the change in raw data, updates are applied to cube data.
  - Scheduled processing initiated every night or several times during the day. This eliminates the need for polling changes or creating database triggers and high system load during the day, which is present in near-real time approach.

SSAS cube settings: aggregations

- By default, SSAS cube does not create and store aggregations
- However, Aggregation Design Wizard can be used to configure whether aggregations should be created (and potentially for which dimensions)
- One of the options is to let Microsoft algorithm decide for which of the dimensions the aggregations should be developed.
- Whether aggregations are created or not will not prevent data cube from working. This can just change the performance.
- Hence, a user is allowed to declare the maximum storage allocation in MBs and observe the impact of extra storage allocation (used for keeping aggregates) on performance gains.

State of the art: Organisational benefits

Organisation benefits of running DWH/OLAP/BI system - summary

Category	Possible benefits
Management and control	1. Constant access to business metrics/KPIs such as volume of sale, divergence from sales target 2. Ability to easily drill down into details when aggregate metrics do not match expectations
Improved business performance	1. Ability to initiate and observe the results of cross-selling 2. Elimination of unsuccessful marketing campaigns
Benefits from running operational BI	1. Improvements in the works of departments using BI tools through improved decision-making and reduced process cost 2. BI can be used also as an embedded BI i.e. some BI-based reports can be integrated into other applications
Process improvement	1. Analysis of process performance through BI can reveal process bottlenecks e.g. most time-consuming steps or steps requiring rework. As an example Boeing company uses near real-time dashboards to track assembly of planes
Improved customer service	1. Identify root causes of warranty issues 2. Observe the number of complaints and their reasons

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zadanie 10 pn. Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych, realizowane w ramach projektu ,,NERW 2 PW. Nauka - Edukacja - Rozwój - Współpraca'', współfinansowanego ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny