Databases II

Datawarehousing
&
Business Intelligence

HO
GENT

What do we learn?

- Basic concepts, advantages and disadvantages of Data Warehousing
- Differences between OLTP and Data Warehousing systems
- Architecture and most important components of a Data Warehouse
- How Data Warehousing has evolved
- Problems with Data Warehousing
- The concept of a data mart
- Activities to start a Data Warehouse project
- Kimball's methodology
- Concepts of Dimensional modelling
- Creation of a DWH

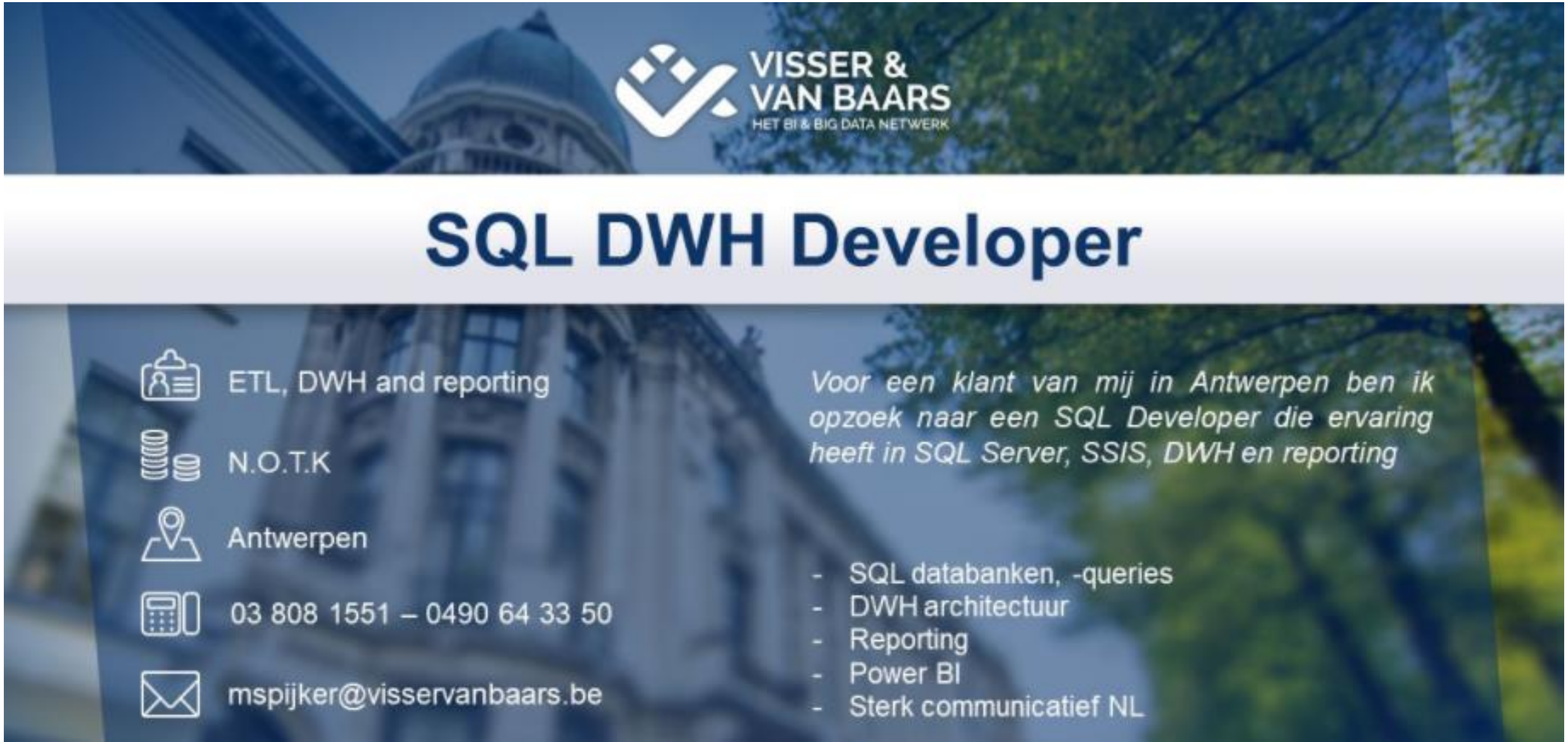
HO
GENT

Introduction

“We proberen continu op de hoogte te blijven. ***Rapportering is daarbij zeer belangrijk.*** ***Ik weet elke dag hoeveel volk er in Pairi Daiza en SnowWorld is geweest, hoeveel dat is in vergelijking met het jaar ervoor, en hoeveel de mensen gespendeerd hebben.*** Ik weet ook elke dag alles van elk van onze deelelementen in Durbuy, ***tot en met hoeveel mensen er snoepjes kopen of mountainbikes gehuurd hebben.*** Wij worden geroepen op speciale momenten, zoals een overname, een investering of personeelskeuzes. Dan willen wij ook toegevoegde waarde bieden, en **alleen als je continu al je dossiers beheerst, kan je er heel snel op inspelen.**"

Marc Coucke in Trends op 22/01/2020

April 2021 Job opening:



VISSER & VAN BAARS
HET BI & BIG DATA NETWERK

SQL DWH Developer

ETL, DWH and reporting

N.O.T.K

Antwerpen

03 808 1551 – 0490 64 33 50

mspijker@visservanbaars.be

Voor een klant van mij in Antwerpen ben ik opzoek naar een SQL Developer die ervaring heeft in SQL Server, SSIS, DWH en reporting

- SQL databanken, -queries
- DWH architectuur
- Reporting
- Power BI
- Sterk communicatief NL

Introduction

There is a growing need to turn data into information

- Flexible business reporting available for the business
- Data in DWH grows exponentially
    - Terabytes of data in a DWH are 'normal'
- Applications using data have become more complex
    - Traditional reporting
    - Advanced analysis
- All traditional DBMS offer DWH facilities

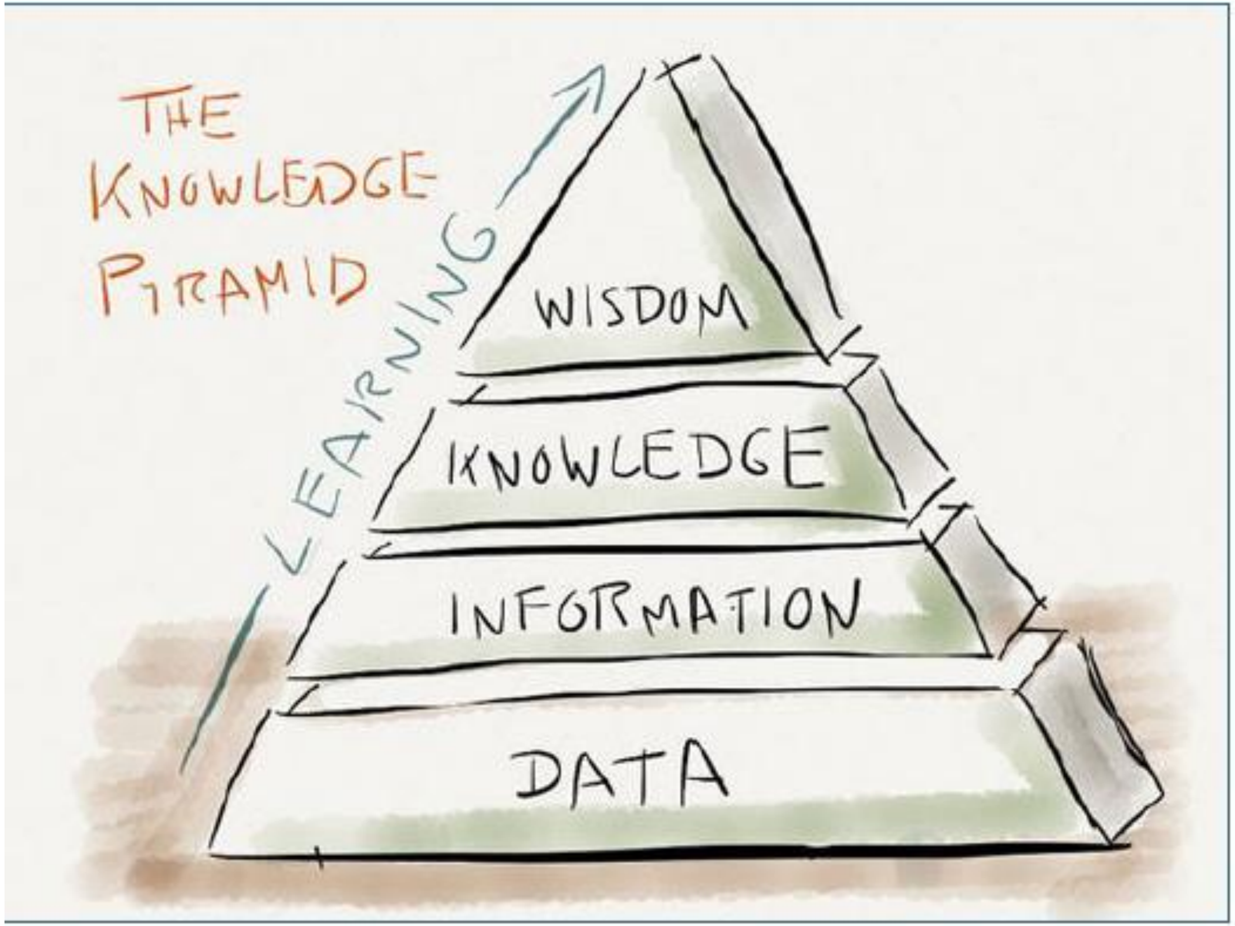


Image: pinterest

HO GENT

Business Intelligence (BI): definition

Business Intelligence(BI) comprises the set of strategies, processes, applications, data, technologies and technical architectures which are used by enterprises to support the collection, data analysis, presentation and dissemination of business information.

BI technologies provide historical, current and predictive views of business operations.

Common functions of business intelligence technologies include reporting, online analytical processing, analytics, data mining, process mining, complex event processing, business performance management, benchmarking, text mining, predictive analytics and prescriptive analytics.

Source: wikipedia

HO
GENT

Business Intelligence (BI): another definition

Business intelligence, or BI, is a blanket term for technology that helps companies organize and query their data to help them improve operations and make more money.

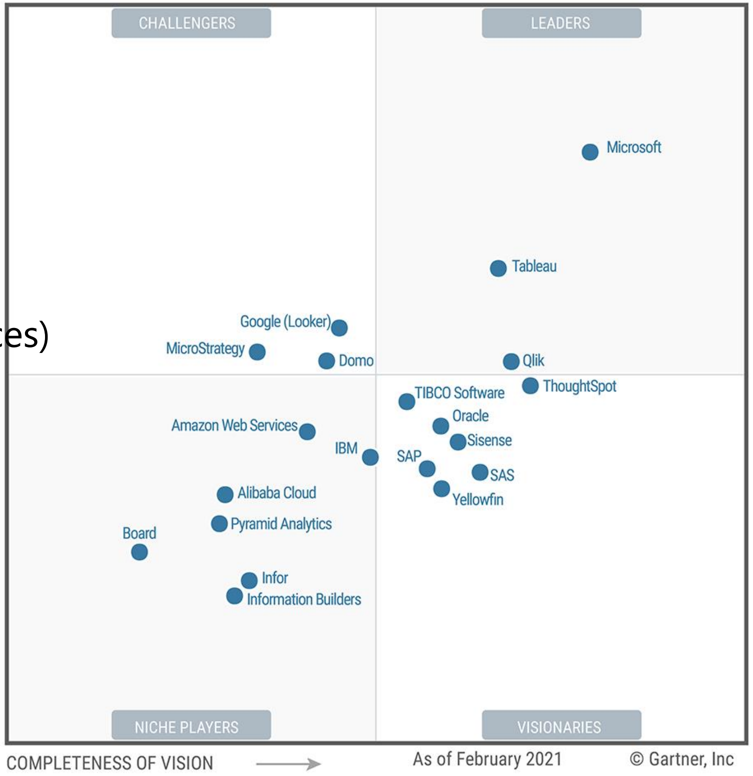
Source: http://fortune.com/2017/04/25/infor-buys-business-intelligence-player

Drivers for increasing use of BI

- Digitalisation (ERP, CRM, PLM, DAM, PIM, …)
- Connectors between BI software and Business software
  - Way of working can stay the same
  - BI comes on top of the existing SW
- Data overflow
- Complexity and Speed of change in the Business Environment
  - Gut feeling and experience is often not sufficient anymore
- Reduce inefficiencies, inaccuracies
  - Anti –Excel culture
- Decreasing Cost

BI technology vendors

- Microsoft:
  - Reporting: Microsoft reporting services + PowerBI
  - BI + data mining: SSAS (SQL Server Analysis Services)
  - ETL: SSIS (SQL Server Integration Services)
- Cognos (now IBM)
  - ETL
  - Reporting tools
- Business Objects (now SAP)
  - Reporting
  - ETL
- SAP Business Warehouse
  - Kubus
- Tableau
- Datastage ETL
- QlikView rapportering



Gartner Magic Quadrant for Business Intelligence and Analytics Platforms, February 2021 (commercial software)

Datawarehouse: definition

A **data warehouse** is an **integrated**, **subject oriented**, **time variant** and **non volatile** collection of data to support decisions taken on management level.

- **Subject oriented**
  - The warehouse is organized around the major subjects of the enterprise (e.g. customers, products, and sales) rather than the major application areas (e.g. customer invoicing, stock control, and product sales).
  - This is reflected in the need to store decision-support data rather than application-oriented data.

HO GENT

Properties

- **Integrated**
  - The data warehouse integrates corporate application-oriented data from different source systems, which often includes data that is inconsistent.
- **Time**
  - Data in the warehouse is only accurate and valid at some point in time or over some time interval.
  - Time-variance is also shown in the extended time that the data is held, the implicit or explicit association of time with all data, and the fact that the data represents a series of snapshots:
    - E.g. unit price of a product can be stored historically or can be a snapshot;
    - DW will built history by taking regular snapshots.

Properties

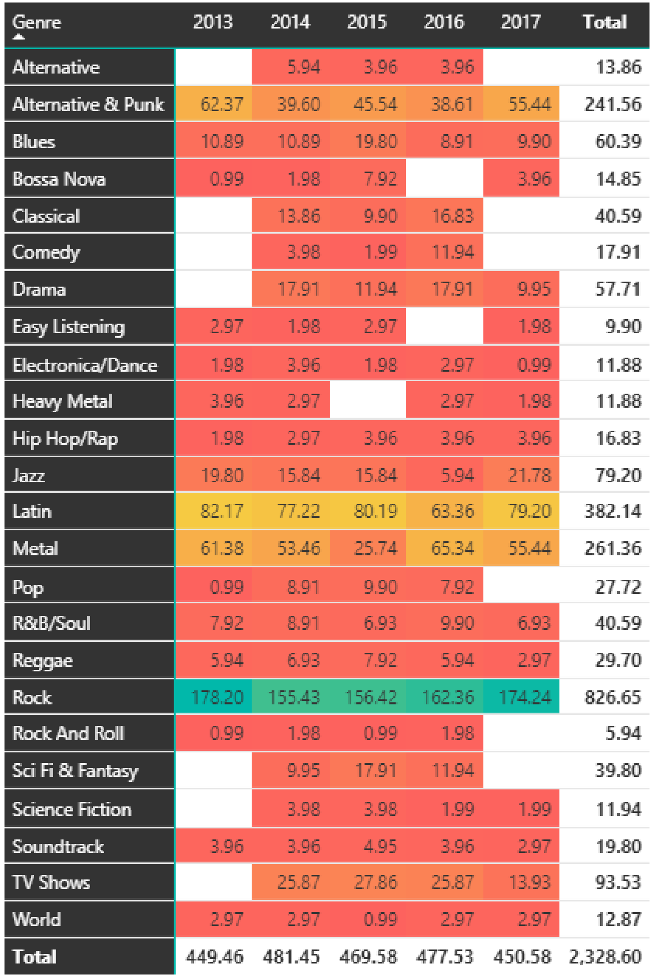
- **Non volatile**
  - Data in the warehouse is not normally updated in real-time (RT) but is refreshed from operational systems on a regular basis. (However, emerging trend is towards RT or near RT DWs).
  - New data is always added as a supplement to the database, rather than a replacement.
- **Aggregated data**
  - E.g. generated by GROUP BY

Goals of DWH

- Reporting
- Analysis of events in the past or actual events
- Predictions based on trend analysis
- Multidimensional reporting
- Empowerment of end user by offering simplified reporting tools (cf. SQL: only specialist can write SQL)
- Data mining

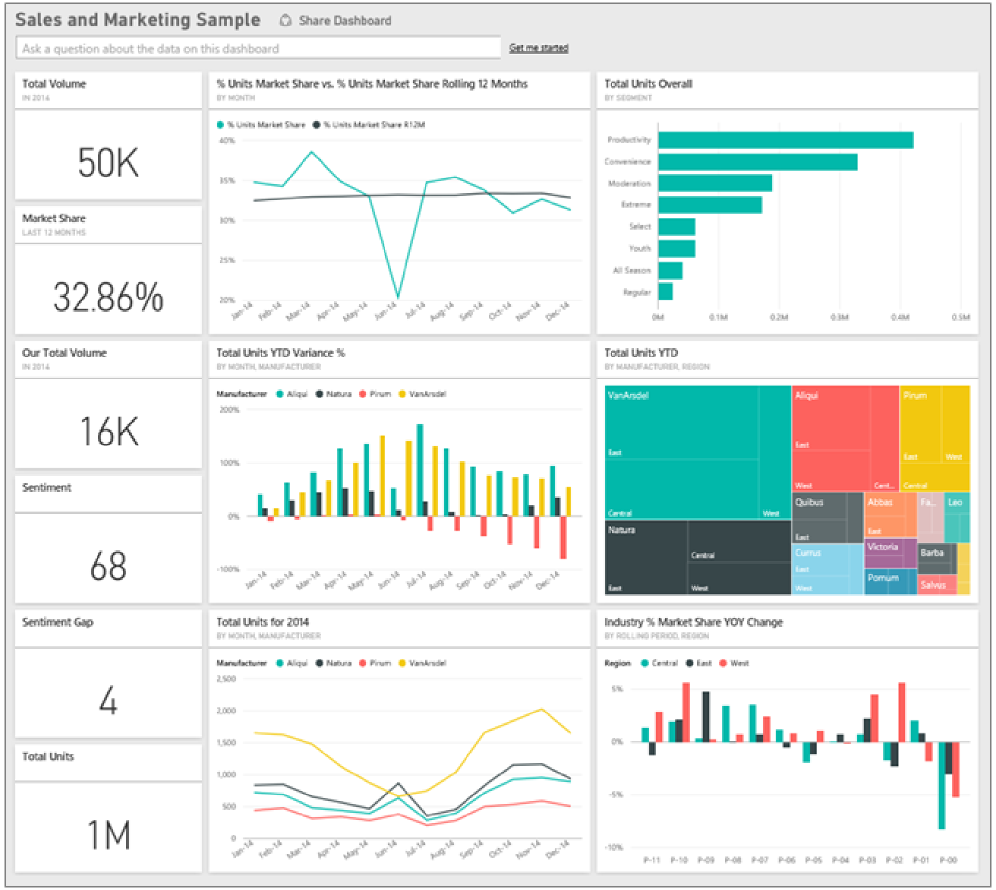
DWH is <u>a</u> technology used to realize BI solutions,
but it is by far <u>not the only</u> technology.

E.g. multidimensional report



Genre	2013	2014	2015	2016	2017	Total
Alternative		5.94	3.96	3.96		13.86
Alternative & Punk	62.37	39.60	45.54	38.61	55.44	241.56
Blues	10.89	10.89	19.80	8.91	9.90	60.39
Bossa Nova	0.99	1.98	7.92		3.96	14.85
Classical		13.86	9.90	16.83		40.59
Comedy		3.98	1.99	11.94		17.91
Drama		17.91	11.94	17.91	9.95	57.71
Easy Listening	2.97	1.98	2.97		1.98	9.90
Electronica/Dance	1.98	3.96	1.98	2.97	0.99	11.88
Heavy Metal	3.96	2.97		2.97	1.98	11.88
Hip Hop/Rap	1.98	2.97	3.96	3.96	3.96	16.83
Jazz	19.80	15.84	15.84	5.94	21.78	79.20
Latin	82.17	77.22	80.19	63.36	79.20	382.14
Metal	61.38	53.46	25.74	65.34	55.44	261.36
Pop	0.99	8.91	9.90	7.92		27.72
R&B/Soul	7.92	8.91	6.93	9.90	6.93	40.59
Reggae	5.94	6.93	7.92	5.94	2.97	29.70
Rock	178.20	155.43	156.42	162.36	174.24	826.65
Rock And Roll	0.99	1.98	0.99	1.98		5.94
Sci Fi & Fantasy		9.95	17.91	11.94		39.80
Science Fiction		3.98	3.98	1.99	1.99	11.94
Soundtrack	3.96	3.96	4.95	3.96	2.97	19.80
TV Shows		25.87	27.86	25.87	13.93	93.53
World	2.97	2.97	0.99	2.97	2.97	12.87
Total	449.46	481.45	469.58	477.53	450.58	2,328.60

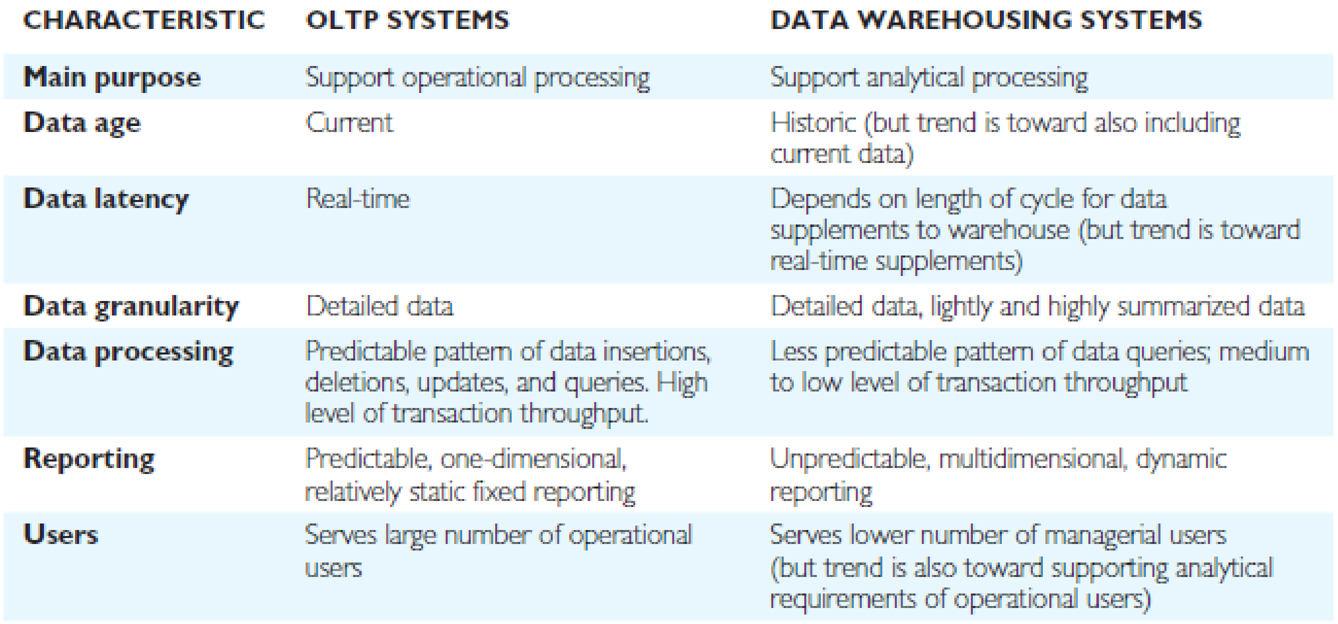
E.g. BI dashboard for manager



Advantages

- **High ROI (return on investment)**
  - Large investments with potentials high ROI after a short period.
- **Competitive advantage**
  - Decision makers get access to data that was not available, unknown or unused before.
- **Increased productivity of corporate decision-makers**
  - Decision makers get one consistent view on the enterprise
    - Because data of different sources can be integrated to one consistent view, that is subject oriented and keeps history.
  - Decision makers can make more substantial, more accurate and more consistent analysis
    - Tools can help turn data into useful information.

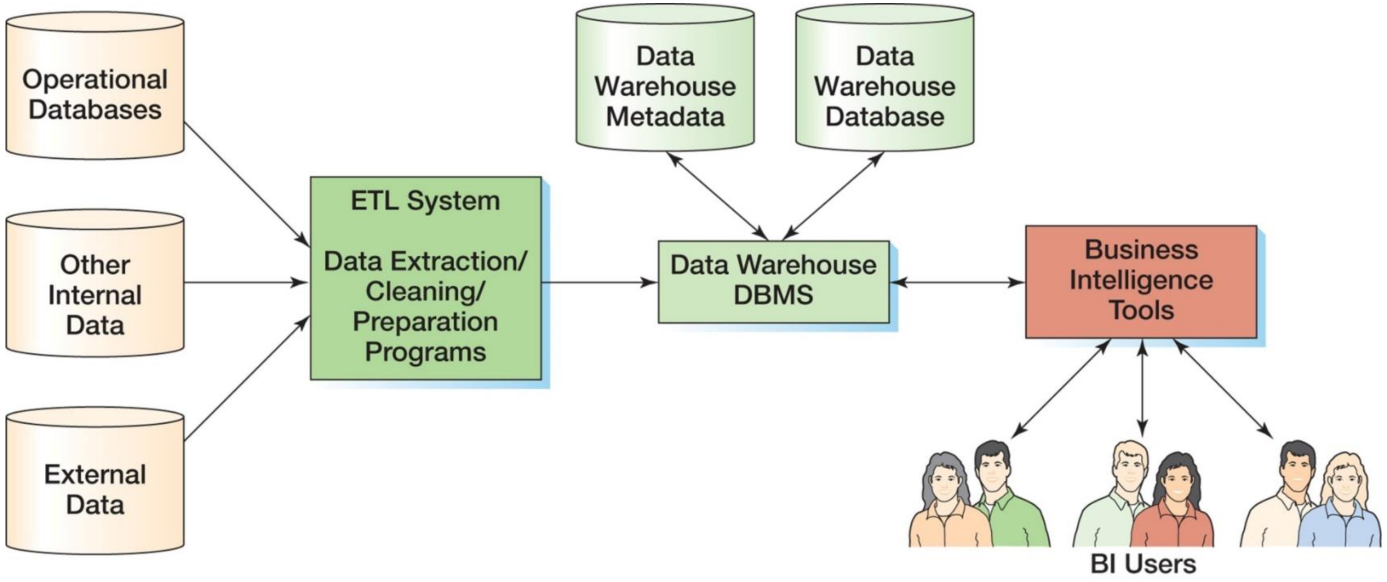
Comparison of OLTP Systems and Data Warehousing



CHARACTERISTIC	OLTP SYSTEMS	DATA WAREHOUSING SYSTEMS
Main purpose	Support operational processing	Support analytical processing
Data age	Current	Historic (but trend is toward also including current data)
Data latency	Real-time	Depends on length of cycle for data supplements to warehouse (but trend is toward real-time supplements)
Data granularity	Detailed data	Detailed data, lightly and highly summarized data
Data processing	Predictable pattern of data insertions, deletions, updates, and queries. High level of transaction throughput.	Less predictable pattern of data queries; medium to low level of transaction throughput
Reporting	Predictable, one-dimensional, relatively static fixed reporting	Unpredictable, multidimensional, dynamic reporting
Users	Serves large number of operational users	Serves lower number of managerial users (but trend is also toward supporting analytical requirements of operational users)

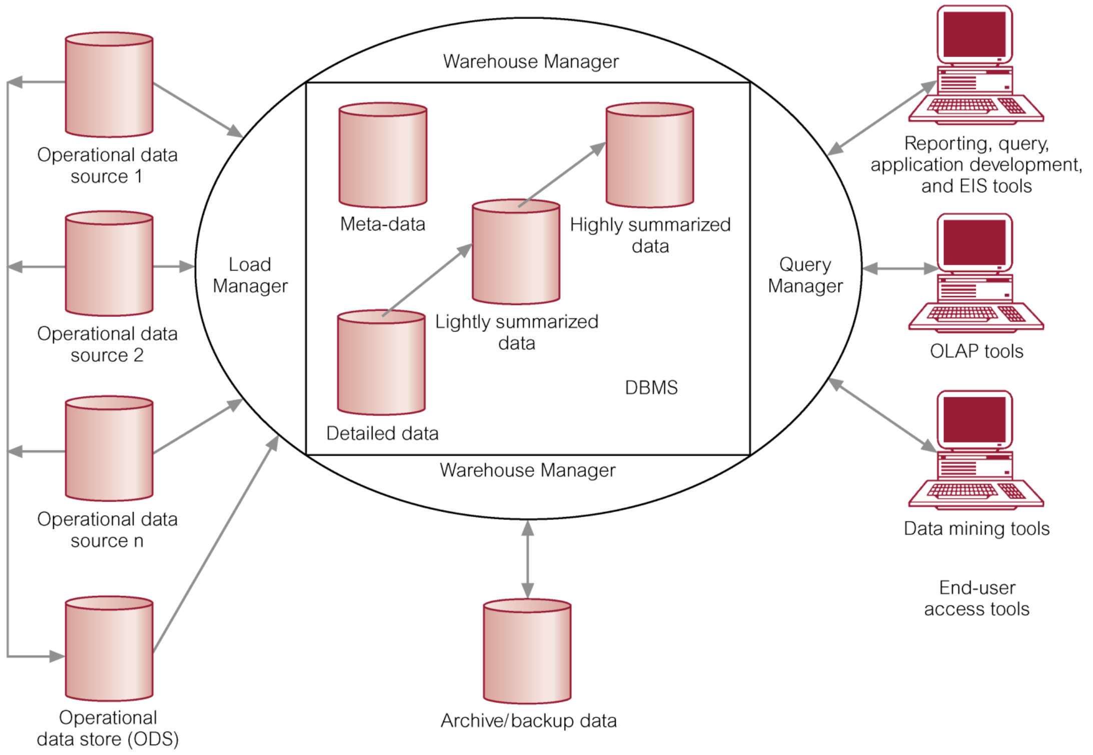
Architecture

DWH components



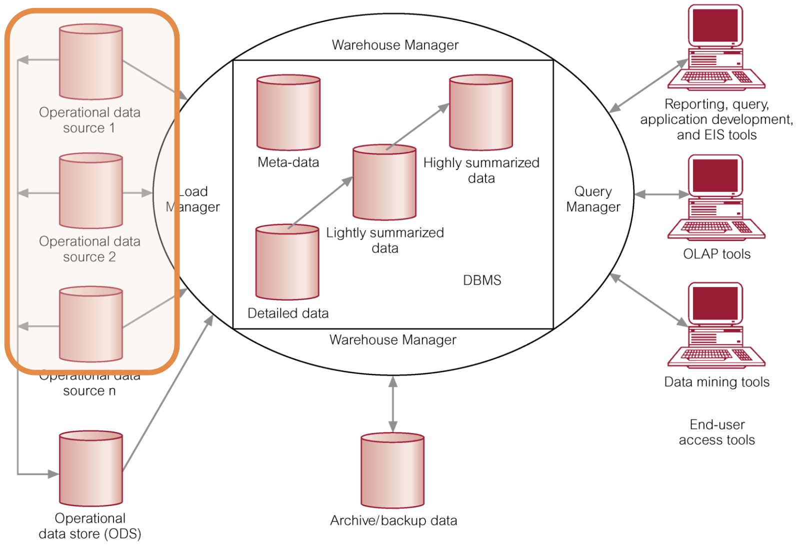
HO GENT

Architecture of a DWH



Architecture of a DWH

- Operational data
- Sources of data
  - Mainframe (hierarchical, network, relational db)
  - Departmental data in files and RDBM systems
  - Private data on workstations and private servers
  - External systems
    - Internet, e.g. prices of competitors
    - Commercial DB
    - Data used by customers or suppliers

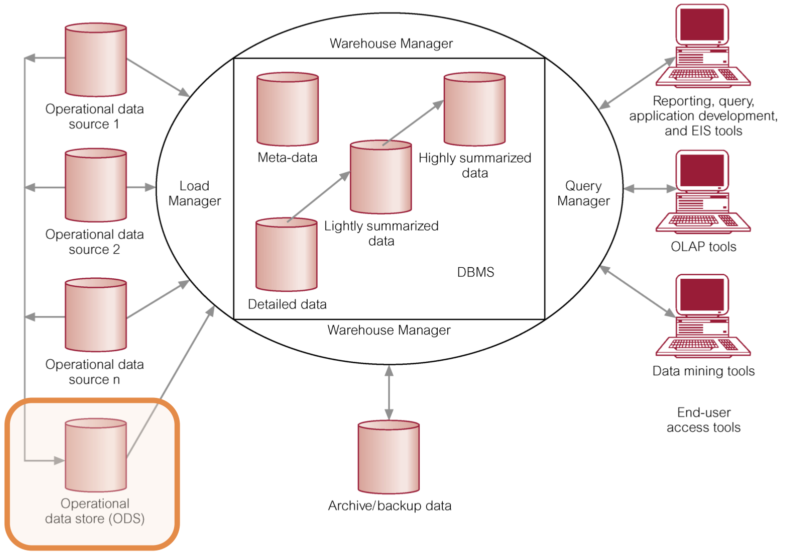


HO GENT

Architecture of a DWH

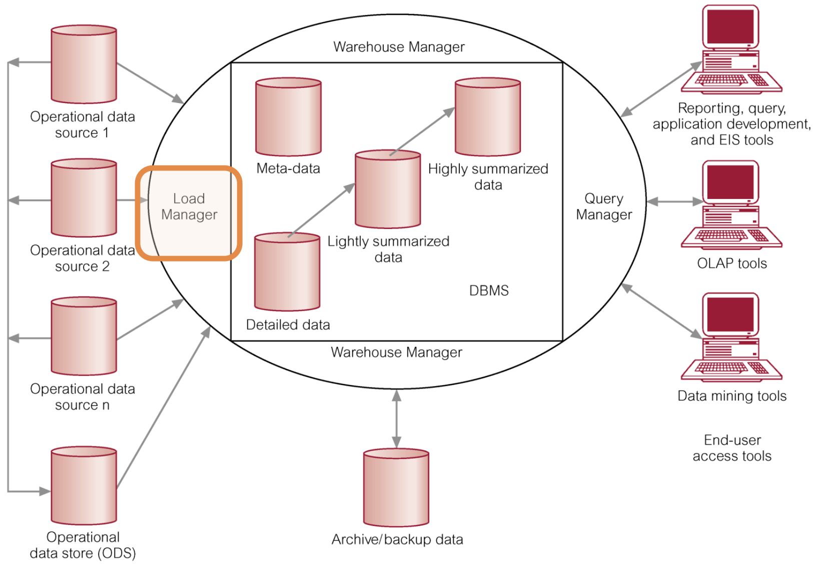
- Operational data source
- Repository
  - Current, integrated data
  - Preparing step in development of DWH,

  or
  - Support for reporting services in legacy systems



Architecture of a DWH

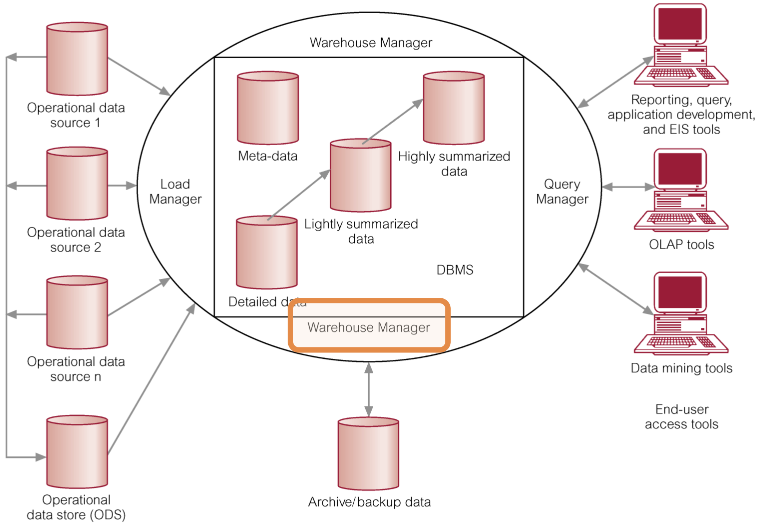
- ETL manager
  - Supports all ETL operations (Extraction, Transformation and Load) on data
    - Either directly on operational data
    - Or on operational data store



HO GENT

Architecture of a DWH

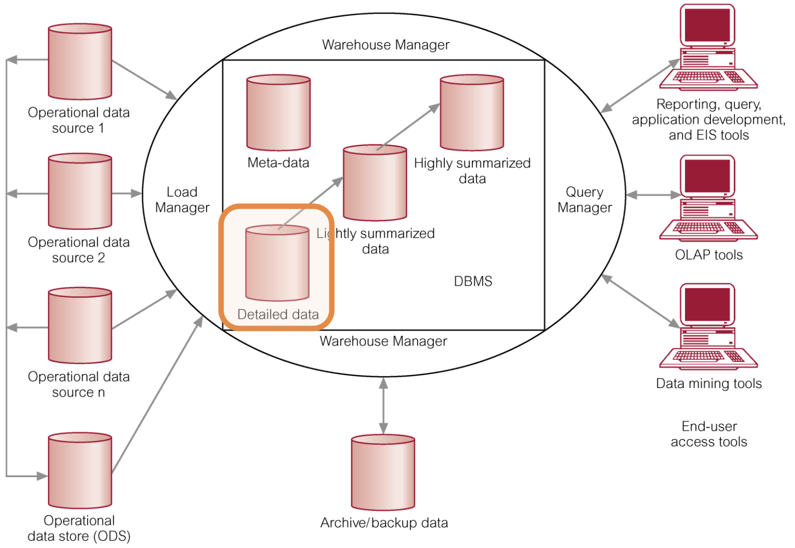
- Warehouse manager
  - Management of data in the DWH
    - Analysis of data to guarantee consistency
    - Transformation and merging data from sources or temporary storage in DWH tables
    - Creation of indexes and views
    - Possibly denormalised
    - Creation of aggregates (data summary)
    - Back-up and archiving of the data



HO GENT

Architecture of a DWH

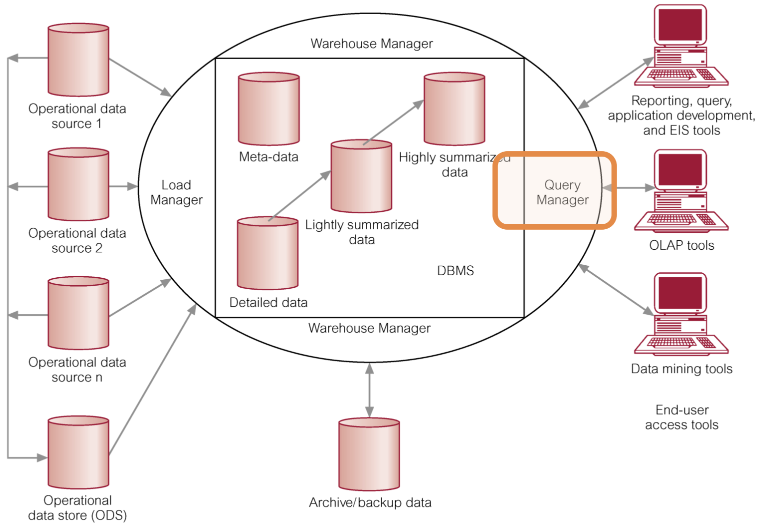
- Detailed data
  - This data is added to the DWH on a regular (e.g. daily) basis



HO GENT

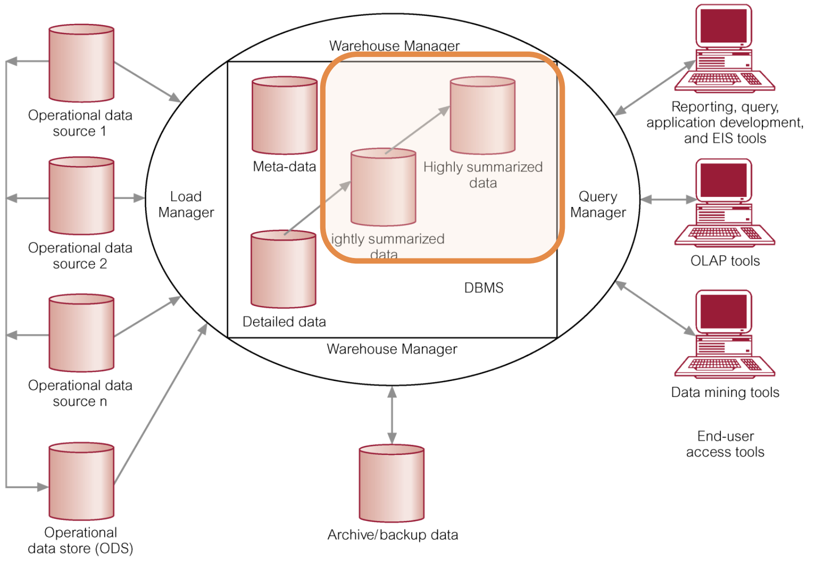
Architecture of a DWH

- Query manager
  - Management of user queries
  - Use of correct tables
  - Execution/scheduling of queries
  - Profile generations
  - Proposals for aggregates and indexes



Architecture of a DWH

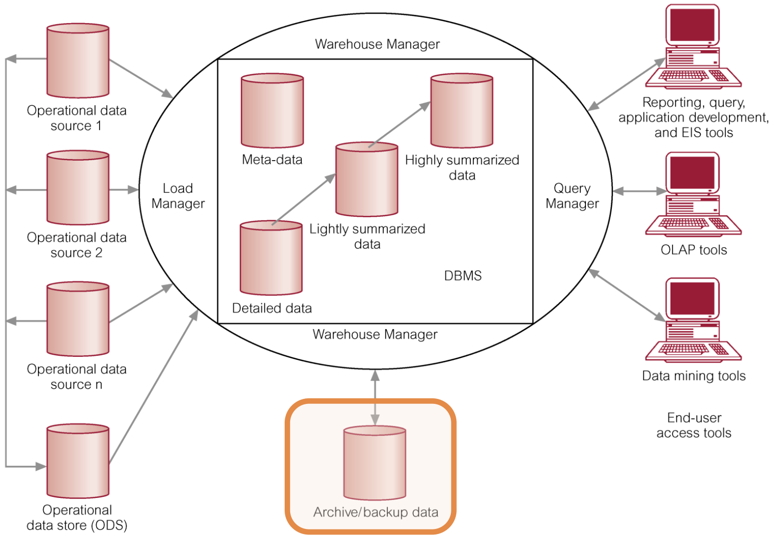
- Summarised data
  - Predefined summarised data
  - Can be adapted in a flexible way so different sorts of queries are supported
  - Improved query execution performance as opposed to detailed data because data is "prepared"



HO GENT

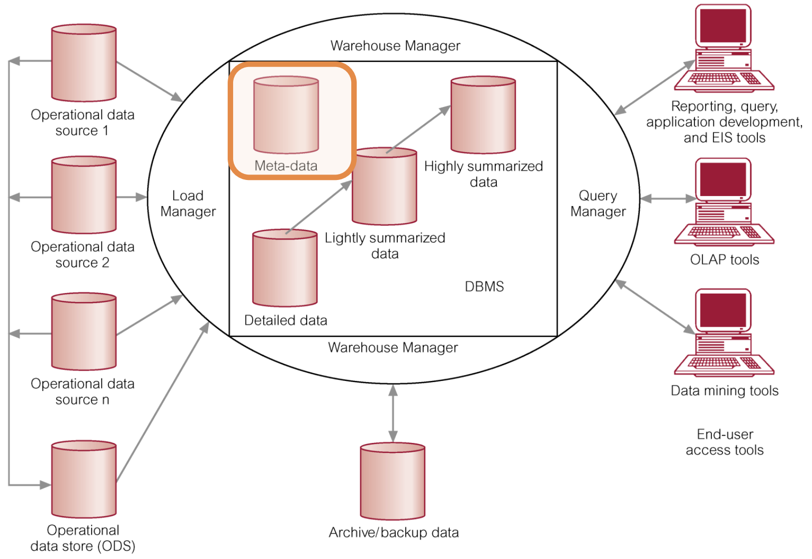
Architecture of a DWH

- Archive/back-up data
  - For both detailed and backup data
    - Summarised data can be kept longer than detailed data



Architecture of a DWH

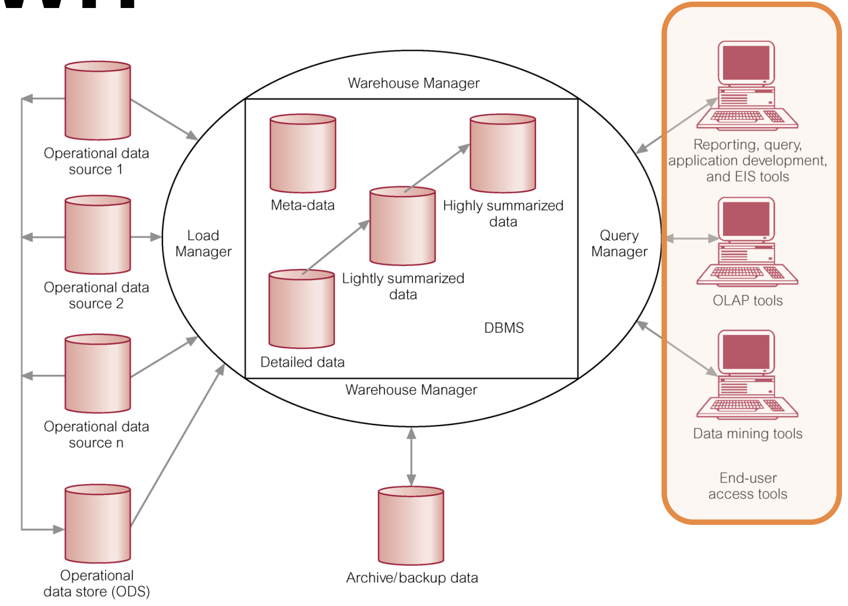
- Meta data
  - Necessary for
    - ETL
    - DWH manager
    - Query manager
- Several versions of metadata each adapted to a specific process
- Allows to determine the source of each data item in the DWH



HO GENT

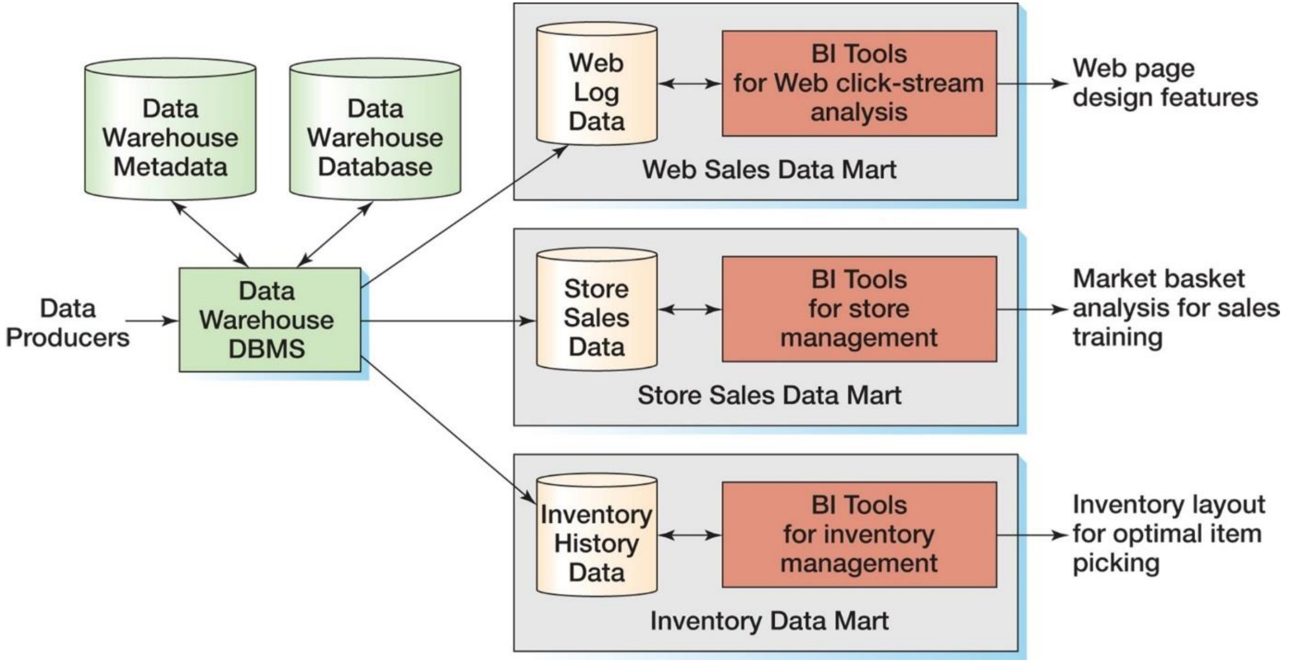
Architecture of a DWH

- End user access tools
  - Reporting and querying
  - Application development tools
  - OLAP tools
  - Data mining tools



HO GENT

DWH & Data marts



Datamart

A DB existing of a **subset of company data** to support the needs of a particular business unit for data analysis, or, to support users sharing the same needs for analyzing **business processes**.

- Why a data mart?
  - To give users access to the data they analyse most frequently
  - To offer data in a way that corresponds to the collective view of a group of users in a department or a group of users in the same business process
  - To improve response time by offering lower data volumes
  - To offer data in a format that fits the tools used by end users (OLAP, datamining tools)
  - Reduction of complexity in the ETL process
  - Reduction of cost as opposed to the setup of a complete DWH

Data Mining Applications

- **Data mining** is used to:
  - Perform what-if analyses
  - Perform predictions ("predictive analysis")
  - Facilitate the decision process
- Data mining applications use sophisticated statistical and mathematical techniques
- Reports are less critical

Problems associated with DWH

- **Underestimation of resources (costs) for ETL**
  - Extraction, transformation and loading of data in the DWH takes a major part of the development time
  - Projects might take years
- **Hidden problems with source systems**
  - Are sometimes only discovered after years
  - Can be solved either in DWH or in operational DB
  - E.g. fields allowing NULL values: some offices never fill out the fields, although data is available and can be useful
- **Required data not captured**
  - Change actual system or develop separate system for those data
  - E.g. the data a customer has registered is not captured
- **Increased end-user demands**
  - Demand for more user friendly, powerful and sophisticated tools
  - Enhanced load of IT personnel
  - Better end user training

Problems associated with DWH

- **Data homogenization**
  - Trying to focus on similarities between data might lower the use of the data
  - E.g. similarities between sale and rent of properties
- **Need for concurrent support of several (historical) versions**
  - Operational systems evolve (i.e. database schema changes over time) but data from older versions and newer versions resides together in the DWH
  - Might be challenging
- **High demand for resources**
  - E.g. disk space
- **Data ownership**
  - Sensitive departmental data (e.g. HR) is highly secured in HR department but is widely available in DWH
- **High maintenance**
  - Each change in business processes or in sources systems influences the DWH (both structure and ETL)

Problems associated with DWH

- **Long projects**
  - Development can take years
  - Phased development through data marts (see further) might be recommended
- **DWH creates expectation of user 'empowering':**
  - Make own reports, analyses
  - Less need for IT
  - But:
    - Meta dictionary that describes data in DWH is necessary
    - Dependency of a few specialists remains
- **Complexity of integration**
  - Different DWH tools have to work together smoothly
- **Complex change and version management**
  - Consistency in reporting between versions of underlying databases
  - **Night might be too short for ETL**

Problems with Operational data

- Dirty data
- Missing values
- Inconsistent data
- Data not integrated
- Wrong format
  - Too fine
  - Not fine enough
- Too much data
  - Too many attributes
  - Too much volume

Design

Design of a DWH

- 2 development methodologies
- **Inmon**
  - Creation of a data model based on all data of the organisation;
  - Enterprise Data Warehouse (EDW);
  - Used to distil data marts for each department;
  - Uses traditional methods for describing EDW:
    - ERD;
    - Tables in normal form.
- **Kimball**
  - Starts by identifying the information requirements (referred to as analytical themes) and associated business processes of the enterprise
    → Data Warehouse Bus Matrix;
  - This first data mart is critical in setting the scene for the later integration of other data marts as they come online;
  - The integration of data marts ultimately leads to the development of an EDW;
  - Uses dimensionality modelling to establish the data model (called star schema) for each data mart.

Kimball's Business Dimensional Lifecycle

- Guiding principle
  - Meet the information requirements of the enterprise by building
    - single,
    - integrated,
    - easy-to-use,
    - high-performance

    information infrastructure, which is delivered in meaningful increments of 6 to 12 months timeframes.
- Goal
  - Deliver the entire solution including
    - the data warehouse,
    - *ad hoc* query tools,
    - reporting applications,
    - advanced analytics
    - all the necessary training and support for the users.

Data Warehouse Sample database

- **Adventureworks Database**
  - Sample Database used by many books and samples about MS SQL Server
  - Exists in an OLTP and DW version
  - Mainly stores data about Products and Sales
  - Download backup files from https://github.com/Microsoft/sql-server-samples/releases/tag/adventureworks

Data Warehouse Sample database

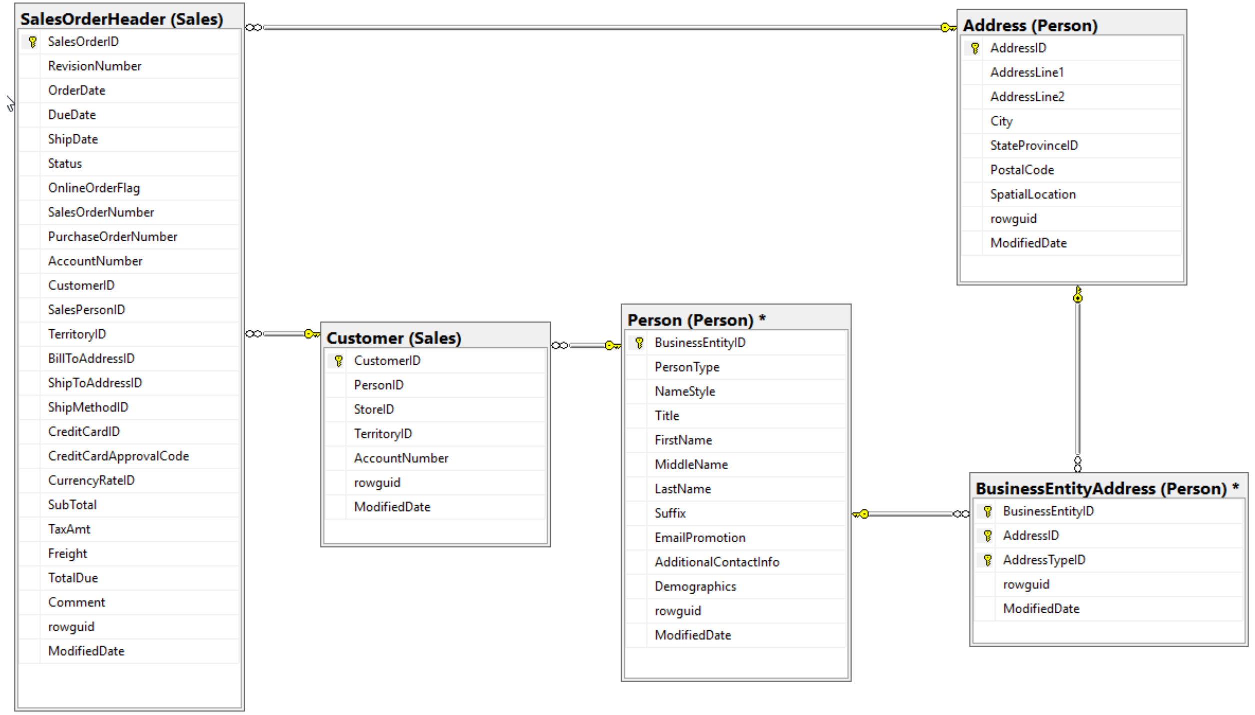
Data Warehouse Sample Queries

- What was the total revenue for UK in the third quarter of 2010?
- What was the total revenue for bike sales for each type of bike (mountain or road) in Germany in 2011?
- What are the three most popular bike types in 2013 and how does this compare with the figures for the previous two years?
- What is the monthly revenue for clothing in each region, compared with rolling 12-monthly prior figures?
- What is the relationship between the total annual revenue for bikes and the average temperature for each country?

Dimensionality modelling: example

ERD AdventureWorks

Often data in an OLTP system is highly normalized, which causes you might need all these tables for a simple sales report:

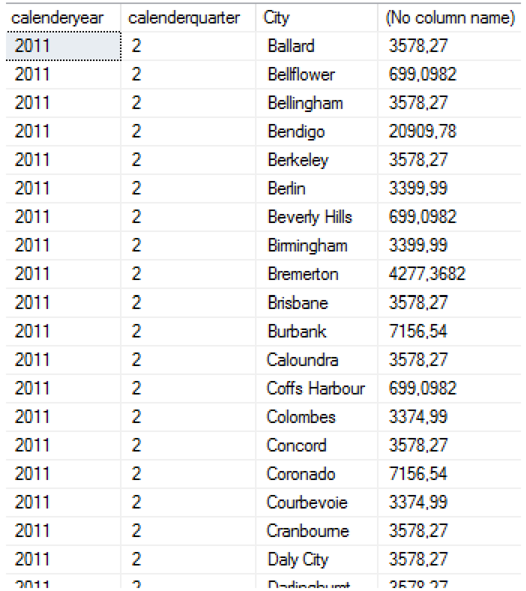


Dimensionality modelling: example

ERD AdventureWorks

The query for a simple sales report might look like this (total salesamount per year, quarter and city):

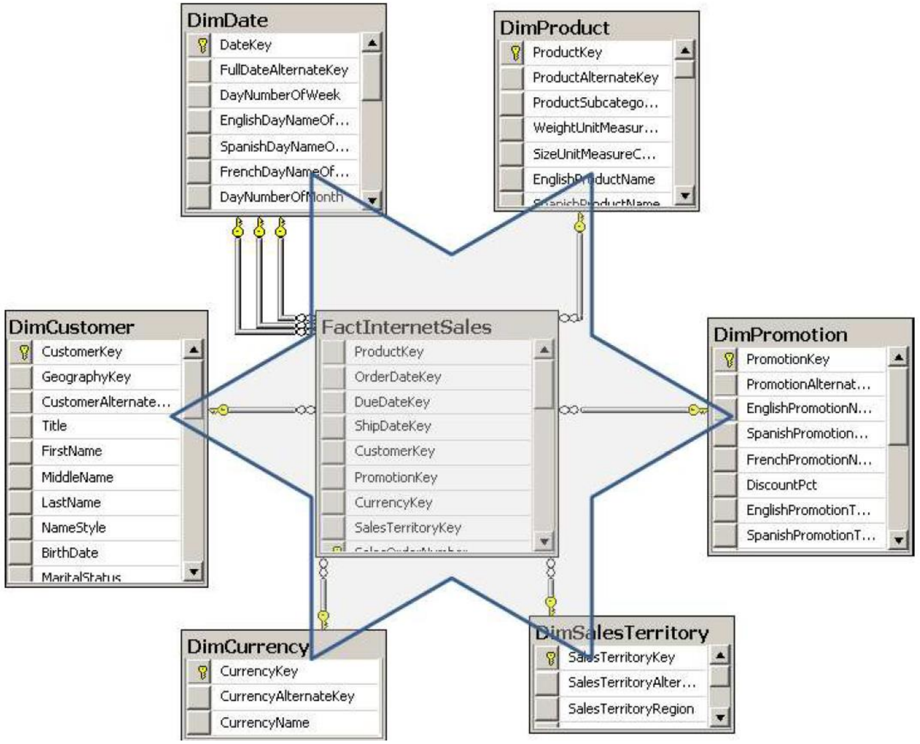
```
select year(orderdate) as calenderyear,
   (month(orderdate)-1)/3+1 as calenderquarter,
    pa.City, sum(subtotal)
from sales.salesorderheader s
join sales.customer c
      on s.CustomerID = c.CustomerID
join person.person p
      on c.PersonID = p.BusinessEntityID
join person.BusinessEntityAddress a
      on p.BusinessEntityID = a.BusinessEntityID
join person.address pa
      on a.AddressID = pa.AddressID
group by year(orderdate), (month(orderdate)-1)/3+1, pa.City
order by 1,2,3;
```



calenderyear	calenderquarter	City	(No column name)
2011	2	Ballard	3578,27
2011	2	Bellflower	699,0982
2011	2	Bellingham	3578,27
2011	2	Bendigo	20909,78
2011	2	Berkeley	3578,27
2011	2	Berlin	3399,99
2011	2	Beverly Hills	699,0982
2011	2	Birmingham	3399,99
2011	2	Bremerton	4277,3682
2011	2	Brisbane	3578,27
2011	2	Burbank	7156,54
2011	2	Caloundra	3578,27
2011	2	Coffs Harbour	699,0982
2011	2	Colombes	3374,99
2011	2	Concord	3578,27
2011	2	Coronado	7156,54
2011	2	Courbevoie	3374,99
2011	2	Cranbourne	3578,27
2011	2	Daly City	3578,27

Star schema

A Star schema is a logical structure that has **a fact table** (containing factual data) in the center, **surrounded by denormalized dimension tables** (containing reference data).



Remark:

- Natural keys from the operational system are available but not as a key in the star schema
- Surrogate integer keys are used instead because they are simpler and faster
- This way independence between OLTP and DWH is ensured

Star schema

To get the same results on the star schema the query looks like this:

```
SELECT calendaryear, calendarquarter, City, sum(salesamount)
FROM factinternetsales AS f
     JOIN dimdate AS d ON f.orderdatekey = d.datekey
     JOIN dimcustomer AS c ON f.CustomerKey = c.CustomerKey
     JOIN dimgeography AS g ON c.GeographyKey = g.GeographyKey
GROUP BY calendaryear, calendarquarter, city
ORDER BY calendaryear, calendarquarter, city;
```

Conclusion: the query on the OLTP system

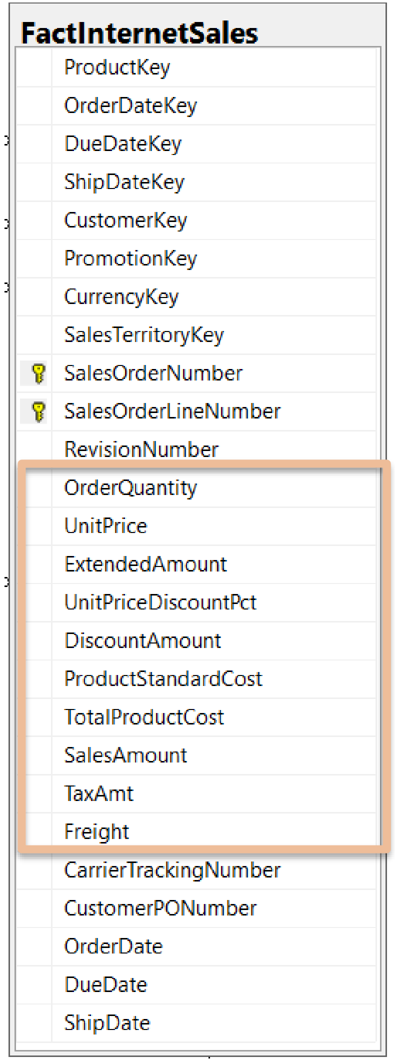
- has deeper join constructs
- joins more tables
- performs calculations (e.g. to get the quarter)

Star schema

- **Fact table** contains data about facts
  - E.g. factual data about sales of property: sales price, commission percentage, …
- **Dimension table** contains reference information
  - E.g. property data (address, etc), buyer, owner, …
- Facts are generated by events that happened (e.g. a sale)
- Most probably facts never change

The fact table

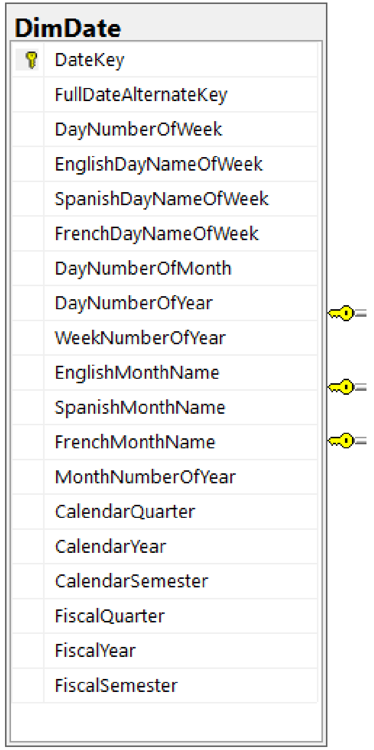
- Bulk of data in the data warehouse is in fact tables, which can be extremely large.
- Important to treat fact data as read-only reference data that will not change over time.
- Most useful fact tables contain one or more numerical measures, or **'facts'** that occur for each record and are numeric and additive.


FactInternetSales

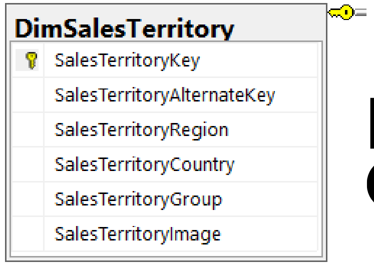
- ProductKey
- OrderDateKey
- DueDateKey
- ShipDateKey
- CustomerKey
- PromotionKey
- CurrencyKey
- SalesTerritoryKey
- 🔑 SalesOrderNumber
- 🔑 SalesOrderLineNumber
- RevisionNumber
- OrderQuantity
- UnitPrice
- ExtendedAmount
- UnitPriceDiscountPct
- DiscountAmount
- ProductStandardCost
- TotalProductCost
- SalesAmount
- TaxAmt
- Freight
- CarrierTrackingNumber
- CustomerPONumber
- OrderDate
- DueDate
- ShipDate

The dimension tables

- Dimension tables usually contain descriptive textual information.
- Dimension attributes are used as the constraints in data warehouse queries. (e.g. in where or having clauses).
- Star schemas can be used to speed up query performance by denormalising reference information into a single dimension table.



DimDate
- DateKey
- FullDateAlternateKey
- DayNumberOfWeek
- EnglishDayNameOfWeek
- SpanishDayNameOfWeek
- FrenchDayNameOfWeek
- DayNumberOfMonth
- DayNumberOfYear
- WeekNumberOfYear
- EnglishMonthName
- SpanishMonthName
- FrenchMonthName
- MonthNumberOfYear
- CalendarQuarter
- CalendarYear
- CalendarSemester
- FiscalQuarter
- FiscalYear
- FiscalSemester



DimSalesTerritory
- SalesTerritoryKey
- SalesTerritoryAlternateKey
- SalesTerritoryRegion
- SalesTerritoryCountry
- SalesTerritoryGroup
- SalesTerritoryImage

Snowflake schema

Snowflake schema is a variant of the star schema that has a **fact table in the centre**, surrounded by **normalised dimension tables**.

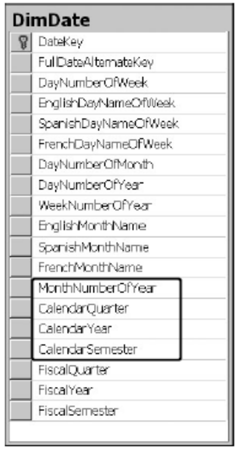


FIGURE 1-4 The *DimDate* denormalized dimension.

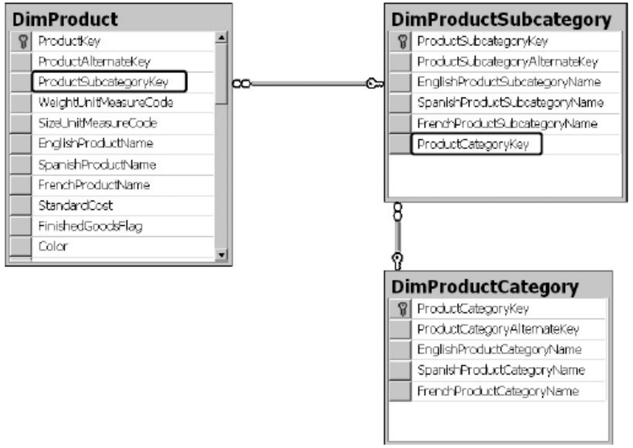


FIGURE 1-5 The *DimProduct* normalized dimension.

Specific Schema Issues

- Surrogate keys
- Granularity of the fact table
- Factless Fact Tables
- Optimizing the dimension tables
- Defining junk dimensions
- Defining outrigger tables
- Slowly changing dimensions
- Rapidly changing dimensions

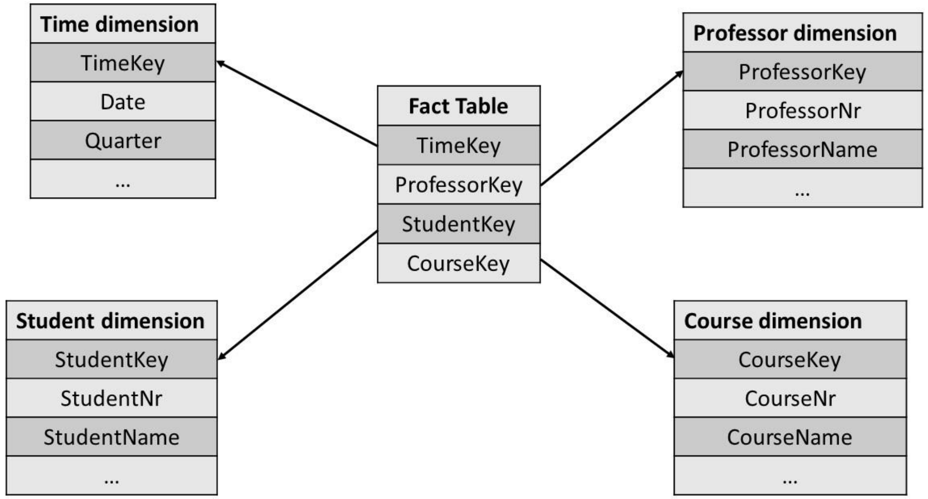
Surrogate keys

- StoreKey, ProductKey, ShipperKey, …
- Meaningless integers
- Cannot use business keys since they usually have a business meaning (so they can be changed, e.g. in the case of a merger)
- Surrogate keys essentially buffer the data warehouse from the operational environment
- Business keys are usually bigger in size
- Business keys are also often re-used over longer periods of time

Granularity of the fact table

- Level of detail of one row of the fact able
- Higher (lower) granularity implies more (fewer) rows
- Trade-off between level of detailed analysis and storage requirements
- Examples:
  - One tuple of the fact table corresponds to one line on a purchase order
  - One tuple of the fact table corresponds to one purchase order
  - One tuple of the fact able corresponds to all purchase orders made by a customer

Factless Fact Tables



This datawarehouse design allows you to answer questions such as:

- Which professor teaches the highest number of courses?
- What is the average number of students that attend a course?
- Which course has the maximum number of students?

Optimizing the dimension tables

- Dimension tables should be heavily indexed to improve query execution time
- On average between 5 and 10 indexes
- E.g. Time
  - TimeKey, Date, DayOfWeek, DayOfMonth, DayOfYear, Month, MonthName, Year, LastDayInWeekFlag, LastDayInMonthFlag, FiscalWeek, HolidayFlag, …

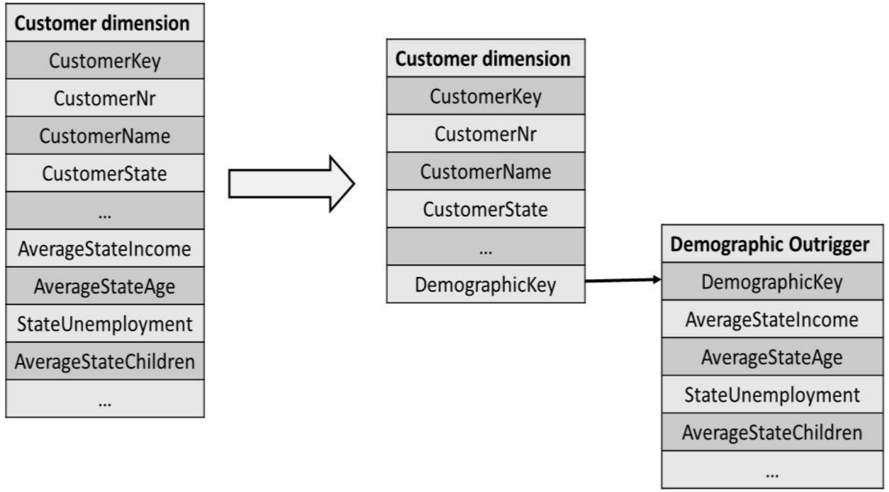
Junk Dimensions

- Deal with low cardinality attribute types such as flags or indicators
- Example: On-line Purchase (Y/N), Payment (cash or credit-card), Discount (Y/N)
- Junk dimension is a dimension that simply enumerates all feasible combinations of values of the low cardinality attribute types
- Advantage?

JunkKey1	On-line purchase	Payment	Discount
1	Yes	Credit-card	Yes
2	Yes	Credit-card	No
3	No	Credit-card	Yes
4	No	Credit-card	No
5	No	Cash	Yes
6	No	Cash	No

Outrigger tables

- Store a set of attribute types of a dimension table which are highly correlated, low in cardinality and updated simultaneously



Slowly Changing Dimensions

- Dimensions that change slowly and irregularly over a period of time
- Example: customer segment ranging from AAA, AA, A, BBB, … to C, determined on a yearly basis

HO
GENT

Slowly Changing Dimensions

- **<u>Approach 1</u>**

<u>OLD STATUS</u>

CustomerKey	CustomerNr	CustomerName	Segment
123456	ABC123	Bart Baesens	AA

<u>NEW STATUS</u>

CustomerKey	CustomerNr	CustomerName	Segment
123456	ABC123	Bart Baesens	AAA

→ Loss of previous state

Slowly Changing Dimensions

- **Approach 2**

OLD STATUS

CustomerKey	CustomerNr	CustomerName	Segment	Start_Date	End_Date	Current_Flag
123456	ABC123	Bart Baesens	AA	27-02-2014	31-12-9999	Y

NEW STATUS

CustomerKey	CustomerNr	CustomerName	Segment	Start_Date	End_Date	Current_Flag
123456	ABC123	Bart Baesens	AA	27-02-2014	27-02-2015	N
123457	ABC123	Bart Baesens	AAA	28-02-2015	31-12-9999	Y

HO
GENT

Slowly Changing Dimensions

- **Approach 3**

OLD STATUS

CustomerKey	CustomerNr	CustomerName	Segment
123456	ABC123	Bart Baesens	AA

NEW STATUS

CustomerKey	CustomerNr	CustomerName	Old Segment	New Segment
123456	ABC123	Bart Baesens	AA	AAA

Slowly Changing Dimensions

- **Approach 4**

Customer

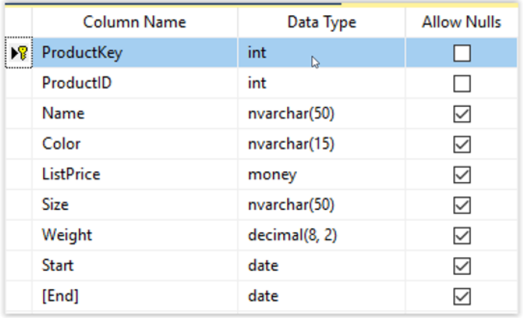
CustomerKey	CustomerNr	CustomerName	Segment
123457	ABC123	Bart Baesens	AAA

Customer_history

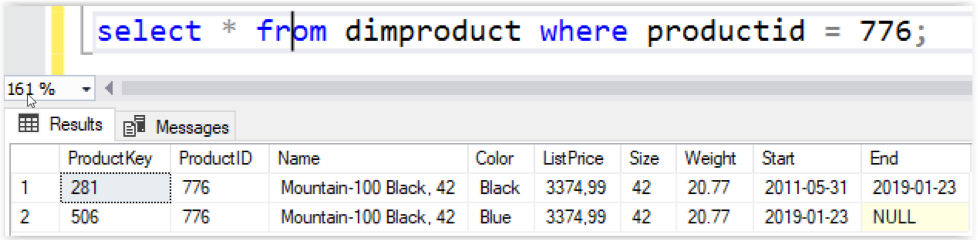
CustomerKey	CustomerNr	CustomerName	Segment	Start_Date	End_Date
123456	ABC123	Bart Baesens	AA	27-02-2014	27-02-2015
123457	ABC123	Bart Baesens	AAA	28-02-2015	31-12-9999

Slowly Changing Dimensions Approach 2 on AdventureworksDW

- Consider DimProduct as a slowly changing dimension
- We want to keep track of old and current values of a.o. Color
- The most common way to handle this is adding two date fields (Start and End) that hold the validity period.
  End = NULL means currently valid.



Column Name	Data Type	Allow Nulls
ProductKey	int	☐
ProductID	int	☐
Name	nvarchar(50)	☑
Color	nvarchar(15)	☑
ListPrice	money	☑
Size	nvarchar(50)	☑
Weight	decimal(8, 2)	☑
Start	date	☑
[End]	date	☑



```
select * from dimproduct where productid = 776;
```

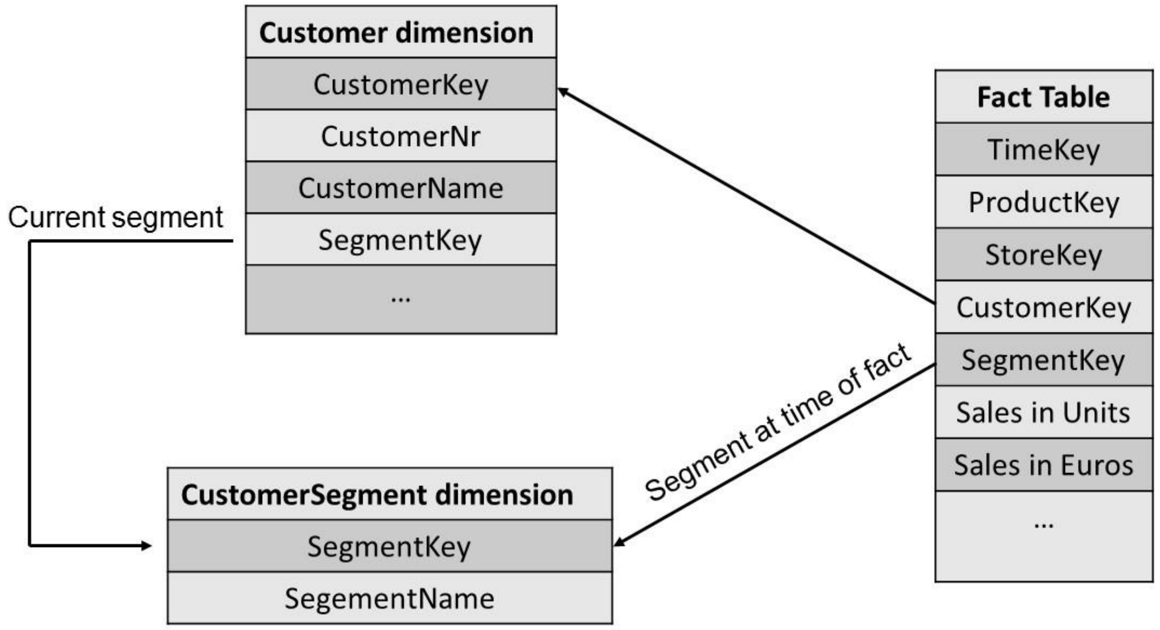
	ProductKey	ProductID	Name	Color	ListPrice	Size	Weight	Start	End
1	281	776	Mountain-100 Black, 42	Black	3374,99	42	20.77	2011-05-31	2019-01-23
2	506	776	Mountain-100 Black, 42	Blue	3374,99	42	20.77	2019-01-23	NULL

Rapidly changing dimensions

- Dimensions that change rapidly and regularly over a period of time
- Example: customer segment ranging from AAA, AA, A, BBB, … to C, determined on a daily basis
- Approaches 2 and 4 discussed in the previous section will result into a lot of rows
- Split customer information into stable (e.g., gender, marital status, …) and rapidly changing information which is put in mini-dimension table

Rapidly changing dimensions

Approach 1:



Rapidly changing dimensions

<u>Approach 1 with sample data</u>:

<u>Fact table</u>

TimeKey	ProductKey	StoreKey	CustomerKey	SegmentKey	Sales in Units	Sales in Euros
200	50	100	1200	1	6	167.94
210	25	130	1400	4	3	54
180	30	150	1000	3	12	384
...						

<u>Customer Dimension</u>

CustomerKey	CustomerNr	CustName	SegmentKey
1000	20	Bart Baesens	2
1200	10	Wilfried Lemahieu	1
1400	5	Seppe vanden Broucke	1

Rapidly changing part

<u>CustomerSegment Dimension</u>

SegmentKey	SegmentName
1	A
2	B
3	C
4	D

Stable part

HO GENT

Advantages of the dimensional model

- Predictable and standard form of the underlying dimensional model offers important advantages:
  - **Efficiency**
    - A consistent DB structure allows tools to have efficient access to the data
  - **Ability to handle changing requirements**
  - The model can easily adapt to changing needs because each dimension is equivalent to the fact table
    - Ideal for ad hoc queries
  - **Extensibility**
    - Adding new facts
    - Adding new dimensions
    - Adding attributes to dimensions
  - **Ability to model common business situations**
  - **Predictable query processing**
    - The way tables are used is predictable (not the queries themselves)

DM and ER models

- **Entity Relationship Diagrams**
  - Used to design the DB of OLTP systems
  - Basics: remove redundancies
    - Redundancy causes update/delete/insert anomalies
  - Ad hoc queries are more difficult
    - Lots of tables can be involved: deep join constructs
- **Dimensional Modelling**
  - Used for design of DWH or data mart
  - Intuitive storage and high performance consulting of data
- A single ER model normally decomposes into multiple DMs.
- Multiple DMs are then associated through 'shared' dimension tables.

Dimensional modelling Stage

- Design issues
  - Choose granularity level:
    - Type 1: the number of dimensions determines the granularity level of the analysis you can get
    - Type 2: every order, summarized by month, quarter, …
  - Duration measures how far back in time the fact table goes
  - Slowly changing dimension problem means that the proper description of the old dimension data must be used with the old fact data.
    - Type 1: the attribute that changes is **overwritten**
    - Type 2: if an attribute changes a **new record** is added to the dimension table
    - Type 3: make sure **old and new value** of the attribute is available in the same record

Exercise

See document

“SSIS sample Project”