



Betriebswirtschaftslehre II

Vorlesung 8: Business Intelligence – Dashboards

Wintersemester 2018/19

Prof. Dr. Martin Schultz

martin.schultz@haw-hamburg.de

Agenda

A faint, artistic background image of a clock face with multiple hands and numbers, partially obscured by the agenda items.

1

Umsetzung der OLAP-Operatoren

2

Informationspräsentation mittels
Dashboards

Inhalte der Vorlesung und Übung

	Termin	Vorlesung	Übung
1	28.09.2018	Einführung und Grundlagen	-
2	05.10.2018	Geschäftsprozessmodellierung	Übung 1 – Gruppe 3/4
3	12.10.2018	Anwendungssysteme in Unternehmen	Übung 1 – Gruppe 1/2
4	19.10.2018	ERP-Systeme	Übung 2 – Gruppe 3/4
5	26.10.2018	ERP-Systeme: ReWe und Einführungsprojekte	Übung 2 – Gruppe 1/2
6	02.11.2018	Business Intelligence - OLAP	Übung 3 – Gruppe 3/4
7	09.11.2018	Business Intelligence - ETL	Übung 3 – Gruppe 1/2
8	16.11.2018	Business Intelligence – Dashboards	Übung 4 – Gruppe 3/4
9	23.11.2018	Informationsmanagement	Übung 4 – Gruppe 1/2
10	30.11.2018	IT-Service-/ Enterprise Architecture-Management	Übung 5 – Gruppe 3/4
11	07.12.2018	IT-Governance/ IT-Compliance	Übung 5 – Gruppe 1/2
12	14.12.2018	Klausurvorbereitung	Übung 6 – Gruppe 3/4
	21.12.2018		Übung 6 – Gruppe 1/2
	11.01.2019		Übung 7 – Gruppe 1/2/3/4

Was sollen Sie mitnehmen...

- Eigenschaften Dashboards für die Management-Unterstützung beschreiben können
- Aufgaben bei der Gestaltung von Dashboards erläutern und bearbeiten können
- Data Mining in den Kontext von Business Intelligence einordnen können

Informationsbereitstellung und Nutzertypen

Spektrum der **Nutzer** von BI-Lösungen ist sehr heterogen (Kenntnisse, Vorlieben), viele unterschiedliche **Darstellungsformen** anwendbar mit unterschiedlichem **Interaktionsgrad** (starr, Verlinkung)

Nutzertyp **Informationskonsument** → **Berichtswesen, Management Support Systeme/ Dashboards**

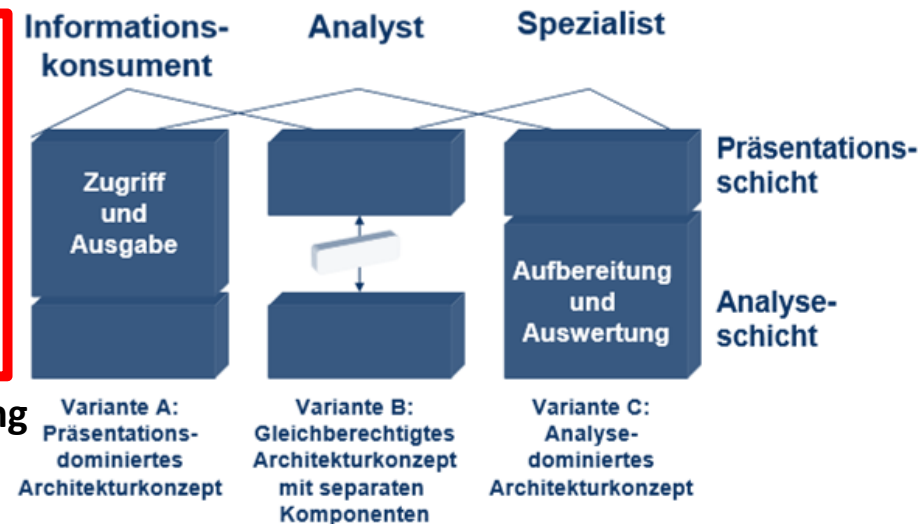
- Nutzertyp der überwiegend Tools verwendet, die das Datenmaterial nach **festen Mustern** aufbereiten und ausgeben

Nutzertyp **Analytiker** → **OLAP**

- Nutzertyp der überwiegend die Funktionalitäten der **navigationsorientierten Analyse** einsetzen und sich frei im Datenbestand bewegen will
- Einfache Methoden und Werkzeuge für Anzeige/ Ausgabe kommen zur Anwendung

Nutzertyp **Spezialist** → **Decision Support, Data Mining**

- Nutzertyp der vorwiegend direkt auf die **methodenorientierten Funktionsbausteine** zurückgreift, um anspruchsvolle Datenanalysen vorzunehmen
- Nimmt dabei funktionale Komplexität und wenig benutzungsfreundliche Oberflächen in Kauf, ggf. Misstrauen gegenüber einfachen Zugriffs- und Ausgabewerkzeugen



(Gluchowski 2008)

Umsetzung der OLAP-Operationen

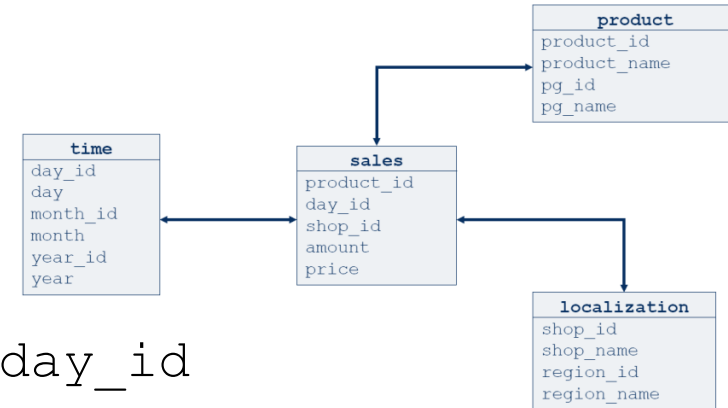
- OLAP Kern-Operationen: **Aggregationen** und Selektionen auf dem **Cube**
 - Roll-up, Drill-Down, Slice, Dice
 - Hierarchische Roll-Ups mit Aggregation auf allen Ebenen (Summe pro Tag, pro Monat, pro Jahr)
 - Aggregation über mehrere Dimensionen (**Cross-Tabs**): Verkäufe pro Marke und Jahr und Shop, Summen in allen Kombinationen
- Anforderungen an Sprachen für OLAP
 - Bereitstellung der notwendigen Operationen
 - **Ökonomisches Design**:
 - Keine vielfach geschachtelten SQL Operationen
 - Keine „Programmierung“
- Hauptsächlich zwei Ansätze:
 - **Erweiterungen von SQL**: Spezielle OLAP-Operationen (ROLLUP, CUBE, ...)
 - **Multidimensional Expressions (MDX)**: Basiert direkt auf die Elemente des multidimensionalen Datenraums: Cube, Dimension, Fakt, ..., Auf dem Vormarsch als de-facto Standard

SQL-Erweiterung für OLAP

- **Annahme:** Star- oder Snowflake Schema
- Triviale Operationen
 - Auswahl (slice, dice): **Joins** und **Selects**
 - Verfeinerung (drill-down): **Joins** und **Selects**
- Einfache Operation
 - Aggregation auf eine Stufe: **Group by**
 - Geht auch in mehreren Dimensionen gleichzeitig
- Nicht völlig trivial
 - Hierarchische Aggregationen
 - Multidimensionale Aggregationen
 - Analytische Funktionen (gleitende Durchschnitte etc.)
 - Alles auch in SQL-92 möglich, aber nur kompliziert auszudrücken und ineffizient in der Bearbeitung

SQL-Erweiterung für OLAP: Standard group by

```
SELECT T.year_id, T.month_id,
       T.day_id, sum(...)
FROM   sales S, product P, time T
WHERE  P.pg_name = „motorbikes“ AND
       T.day_id = S.day_id
GROUP BY T.year_id, T.month_id, T.day_id
```



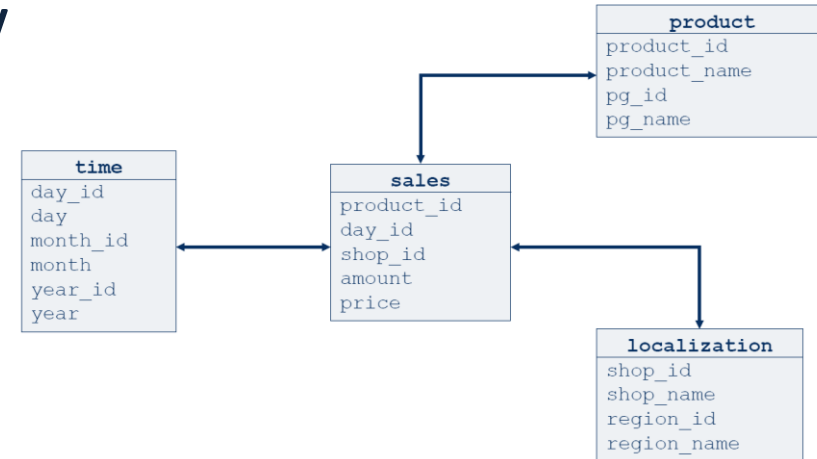
- Summen nur für die Tage
- Keine Summen pro Monat / pro Jahr
- Solche zusätzlichen Summen in einem SFW-Block nicht möglich (SQL92)
→ Unions notwendig

Year	Month	Day	Sum (...)
2013	1	1	150
...
2013	1	31	250
...
2017	1	1	123
...
2017	12	31	45

SQL-Erweiterung für OLAP: Standard group by

- Zwischensummen benötigen **UNION** und eine **Query pro Klassifikationsstufe**

```
SELECT      T.year_id, T.month_id
           T.day_id, sum(...)
FROM    sales S, product P, time T
WHERE   P.pg_name = „motorbikes“ AND
        T.day_id = S.day_id
GROUP BY T.year_id, T.month_id, T.day_id
```



UNION

```
SELECT T.year_id, T.month_id
       T.day_id, sum(...)
FROM   sales S, product P, time T
WHERE  P.pg_name = „motorbikes“ AND
       T.day_id = S.day_id
GROUP BY T.year_id, T.month_id
```

UNION

```
SELECT T.year_id, T.month_id
       T.day_id, sum(...)
FROM   sales S, product P, time T
WHERE  P.pg_name = „motorbikes“ AND
       T.day_id = S.day_id
GROUP BY T.year_id
```

Year	Month	Day	Sum (...)
2013	1	1	150
...
2017	12	31	250
...
2017	1	Null	3.453
...
2017	Null	Null	4.635.291

SQL-Erweiterung für OLAP: ROLLUP

- Herkömmliches SQL für hierarchische ROLLUP Dimension mit k Stufen – Union von k Queries
 - **k Scans der Faktentabelle**
 - Typischerweise keine Optimierung wg. fehlender Multiple-Query Optimierung in den meisten RDBMS
- Ungünstige Ergebnisreihenfolge (schwierig zu sortieren)
- SQL Erweiterung: **ROLLUP** Operator
- Hierarchische Aggregation mit Zwischensummen
 - Summen werden durch „ALL“ als Wert repräsentiert
 - In Oracle ist es eine NULL statt ALL
- Identifizierung über **GROUPING**-Funktion

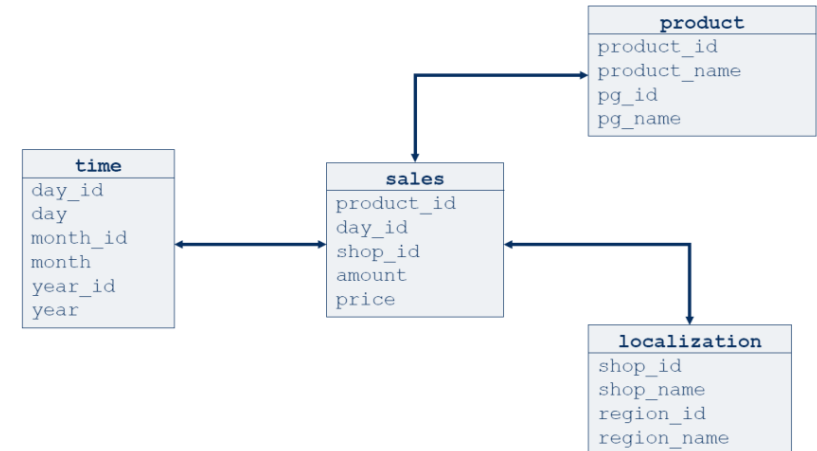
Beispiel:

```
SELECT T.year_id, T.month_id, T.day_id, sum(...)
FROM   sales S, product P, time T
WHERE  T.day_id = S.day_id
GROUP BY ROLLUP(T.year_id, T.month_id, T.day_id)
```

SQL-Erweiterung für OLAP: ROLLUP - Beispiel

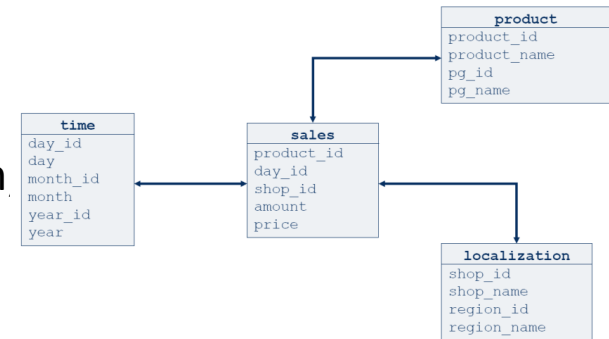
Beispiel: `SELECT T.year_id, T.month_id, T.day_id, sum(...)`
`FROM sales S, product P, time T`
`WHERE T.day_id = S.day_id`
`GROUP BY ROLLUP(T.year_id, T.month_id, T.day_id)`

Year	Month	Day	Sum (...)
2013	Jan	1	150
...
2013	Jan	31	250
2013	Jan	ALL	10.250
2013	Feb	1	350
...
2013	March	31	600
2013	March	ALL	756
...
2013	ALL	ALL	180.056
...
ALL	ALL	ALL	1 Mio



SQL-Erweiterung für OLAP: Bedingtes ROLLUP

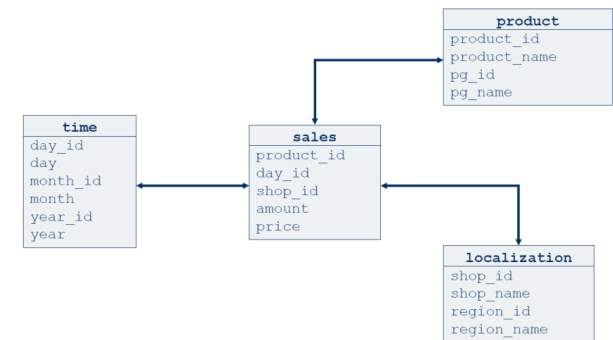
- Man will nicht immer alle Klassifikationsknoten sehen
 - z.B.: Hierarchische Aggregation über Shop und Region aber explizite Ausweisung nur für die Shops „Poppenbüttel“, „Eimsbüttel“, „Eilbek“
 - Gesamtsumme pro Region soll erhalten bleiben
- Selektion in der WHERE Klausel geht nicht, weil sonst die Gesamtsummen pro Region verfälscht werden
- Zwei Möglichkeiten
 - Verwendung von **HAVING**, um die überflüssigen Tupel aus dem Ergebnis zu filtern
 - Erzeugt viele Tupel und wirft sie dann weg
 - Verwendung einer **CASE** Anweisung im ROLLUP Operator Erzeugt nur die gewünschten Tupel
 - Benötigt prinzipiell gleich viele Scans, aber weniger Speicherplatz



SQL-Erweiterung für OLAP: Bedingtes ROLLUP - Beispiel

Beispiel: `SELECT L.shop_id, L.region_id, sum(...)
FROM sales S, localization L
WHERE S.shop_id = L.shop_id
GROUP BY ROLLUP(L.region_id,
CASE WHEN L.shop_id IN ('Eims...', 'Poppen...')
THEN L.shop_id
ELSE 'Others'
End))`

Region_id	Shop_id	Sum (...)
Bayern	Others	150
Bayern	ALL	...
Hamburg	Eimsbüttel	250
Hamburg	Poppenbüttel	1.250
Hamburg	Others	4.500
Hamburg	ALL	10.500
...
ALL	ALL	1 Mio



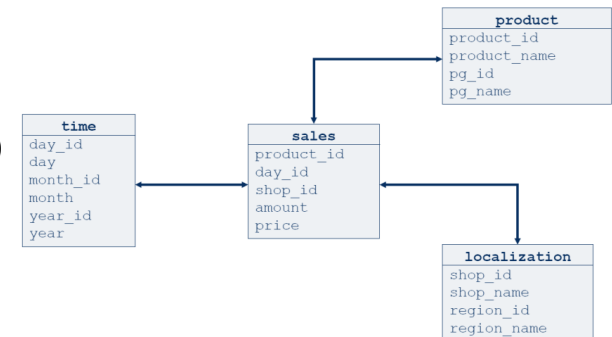
SQL-Erweiterung für OLAP: Grouping

Mit Hilfe des Grouping-Befehls können Summenzeilen identifiziert werden, um sie bzgl. der Ausgabe gesondert zu behandeln

Beispiel:

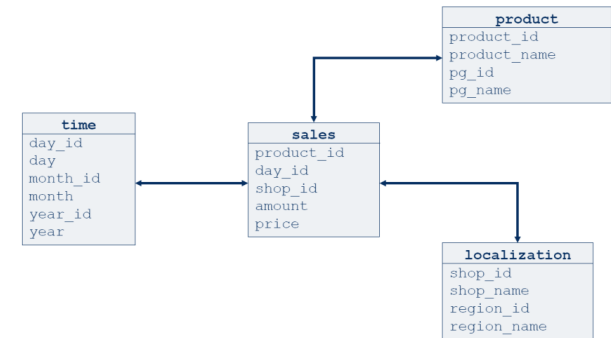
```
SELECT (CASE WHEN GROUPING(L.shop_id) = 1 THEN 'ALL Shops')
      as Shops
      CASE WHEN GROUPING(L.region_id) = 1 THEN 'ALL Regions')
      Regions,
      sum(...)
FROM   sales S, localization L
WHERE  S.shop_id = L.shop_id
GROUP BY ROLLUP(L.region_id, L.shop_id)
```

Regions	Shops	Sum (...)
...
Bayern	München	150
Bayern	ALL Shops	...
Hamburg	Eimsbüttel	250
Hamburg	Poppenbüttel	1.250
...
Hamburg	ALL Shops	10.500
...
ALL Regions	ALL Shops	1 Mio



SQL-Erweiterung für OLAP: CUBE

- **Annahme:** Ein Würfel mit d Dimensionen, mit jeweils eine Klassifikationsstufe
 - Jede Dimension kann in Gruppierung enthalten sein oder nicht
 - 2^d Gruppierungsmöglichkeiten
- **Lösung 1:** Herkömmliches SQL
 - Sehr lange SQL-Statements, viele UNIONS
 - Wahrscheinlich 2^d Scans der Faktentabelle
- **Lösung 2:** SQL Erweiterung mit dem **CUBE Operator**
 - Berechnung der Summen von sämtlichen Kombinationen der Argumente (Klassifikationsstufen)

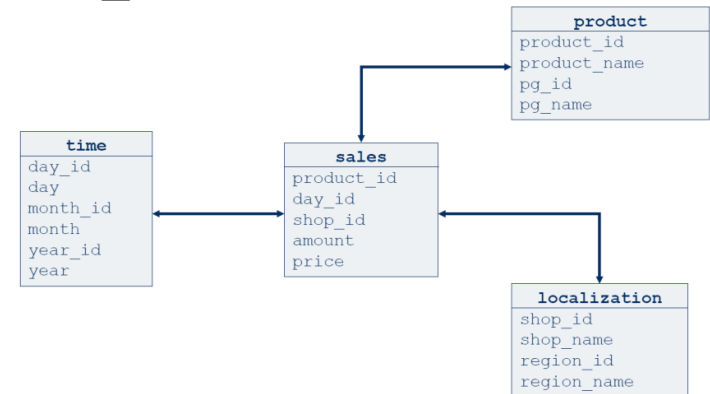


Beispiel: `SELECT pg_id, shop_id, year_id, sum(...)
FROM sales S join product using (product_id) ...
GROUP BY CUBE(pg_id, shop_id, year_id)`

SQL-Erweiterung für OLAP: CUBE - Beispiel

Beispiel: `SELECT pg_id, shop_id, year_id, sum(...)
FROM sales S join product using (product_id) ...
GROUP BY CUBE(pg_id, shop_id, year_id)`

pg_id	shop_id	year	Sum (...)
...
Bier	Eimsbüttel	ALL	...
Bier	Poppenbüttel	ALL	...
Bier	ALL	2013	...
Wein	ALL	2014	...
ALL
...	ALL	ALL	...
ALL	ALL	2013	...
ALL	ALL	2014	...
ALL	Eimsbüttel	ALL	...
ALL	Poppenbüttel	ALL	...
ALL	ALL	ALL	...



Vorberechnete Aggregate: Fact-Constellation

Speicherung vorberechneter Aggregate in Faktentabelle

- Eintrag in der Faktentabelle muss auf geeignete Dimensionswerte verweisen
- Unterscheidung in Dimensionstabelle über spezielle Attribute (Bsp.: „Stufe“)
- *Beispiel:* Anzahl Verkäufe für jeden Tag und jede Filiale pro Produktgruppe

Dimensionstabelle Produkt (Star Schema)

P_ID	P_Bezeichnung	P_PGruppe	P_PKategorie	P_Stufe
142	Merlot	Wein	Getränke	0
143	Chardonnay	Wein	Getränke	0
		...		
100	NULL	Wein	Getränke	1
271	Guinness	Bier	Getränke	0
281	Radler	Bier	Getränke	0
		...		
200	NULL	Bier	Getränke	1
		...		
900	NULL	NULL	Getränke	2
		...		

```
SELECT O_Filiale, YEAR(Z_Datum), SUM(V_Anzahl)
FROM Verkauf, Ort, Produkt, Zeit
WHERE V_Produkt_ID = P_ID AND V_Zeit_ID = Z_ID AND
      V_Ort_ID = O_ID AND
      P_Produktgruppe = 'Wein' AND P_Stufe = 1
GROUP BY O_Filiale, YEAR(Z_Datum)
```



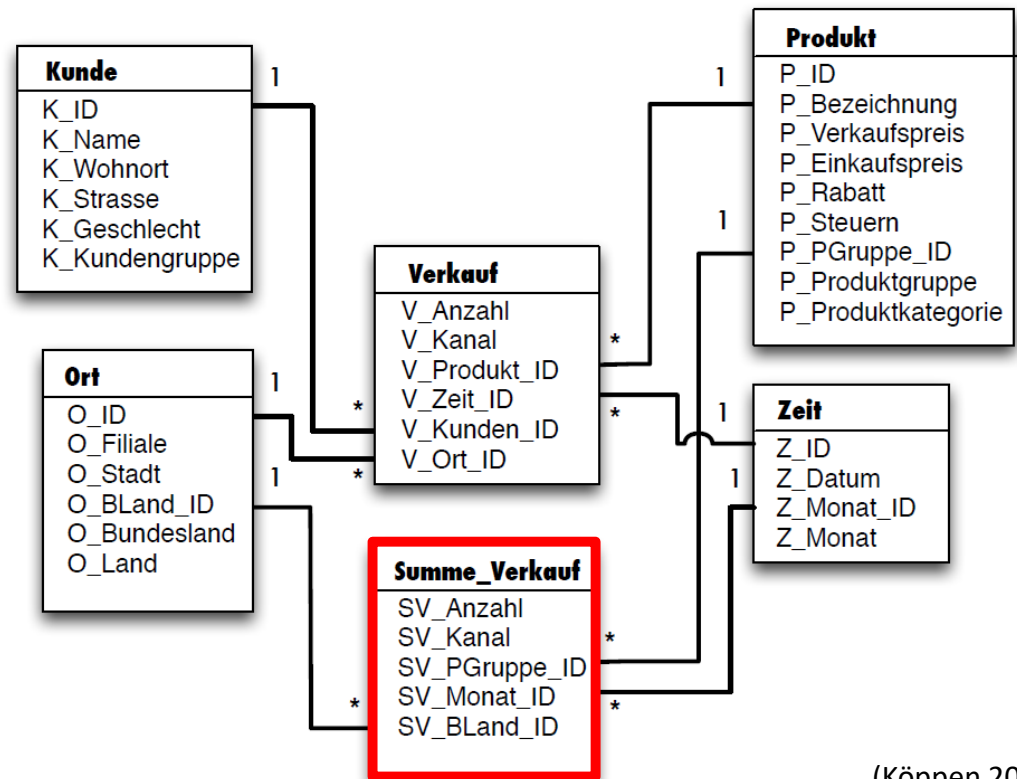
- Reduktion der Treffer in der Faktentabelle
- Vermeidung von Mehrfachzählung

(Köppen 2014)

Vorberechnete Aggregate: Fact-Constellation-Schema

- Auslagerung vorberechnete Aggregate in eigene Faktentabellen (Summentabellen)
- Verweis von der Faktentabelle direkt auf die Attribute der jeweiligen Hierarchieebene der Dimensionstabellen (z.B. SV_BLand_ID)

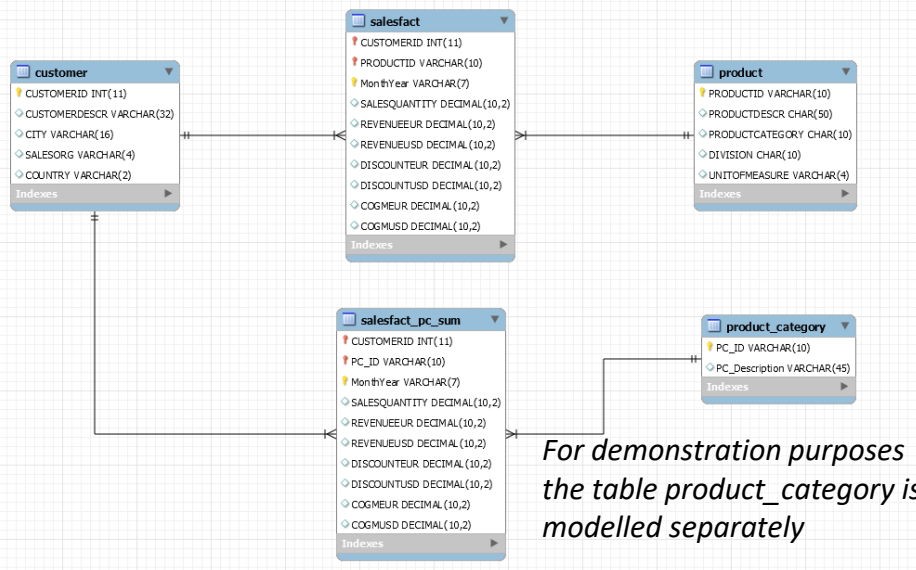
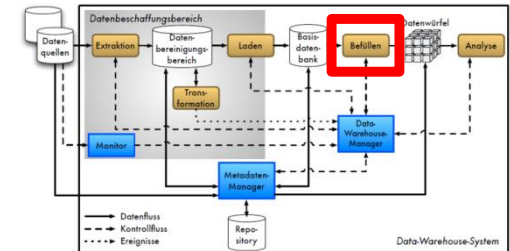
- *Beispiel:* Vorberechnete Aggregate für die Kombination aus
 - Monat (Zeit)
 - Bundesland (Ort)
 - ProduktGruppe (Produkt)
 - alle Kunden (Kunde)



(Köppen 2014)

Filling: Sample Data from the Exercise with pentaho

- Filling of fact constellation table



SQL-Statement for Aggregation

```
SELECT CUSTOMERID,
        PRODUCTCATEGORY,
        EXTRACT(YEAR_MONTH from TransactionDATE) as TransactionDATE,
        sum(SALESQUANTITY),
        sum(REVENUEEUR),
        sum(REVENUEUSD),
        sum(DISCOUNTEUR),
        sum(DISCOUNTUSD),
        sum(COSTOFGOODSEUR),
        sum(COSTOFGOODSUSD)
FROM `bi_sales_basis_database`.`bi_sales_dwh`
group by CUSTOMERID, PRODUCTCATEGORY, EXTRACT(YEAR_MONTH from TransactionDATE)
```

Result Table (salesfact_pc_sum):

6.168 data records

CUSTOMERID	PC_ID	MonthYear	SALESQUANTITY	REVENUEEUR	REVENUEUSD	DISCOUNTEUR	DISCOUNTUSD	COGMEUR	COGMUSD
1000	ACC	200701	15.00	440.00	440.00	15.00	15.00	220.00	220.00
1000	ACC	200702	26.00	798.00	798.00	25.00	25.00	399.00	399.00
1000	ACC	200703	34.00	1160.00	1160.00	37.00	37.00	581.00	581.00
1000	ACC	200704	64.00	2110.00	2110.00	64.00	64.00	1055.00	1055.00
1000	ACC	200705	85.00	3032.00	3032.00	92.00	92.00	1516.00	1516.00
1000	ACC	200706	81.00	2835.00	2835.00	86.00	86.00	1419.00	1419.00

Multidimensional Expressions (MDX)

- Eigene Anfragesprache
 - Sprache ohne feste Semantik (by example)
 - **Konzepte des multidimensionalen Datenmodells** (z.B. measures, dimensions, classifications) sind **direkt in den Anfragen** verwendbar
 - Dimensions = Dimensionen
 - Level = Klassifikationsstufe
 - Multiple hierarchies = Verschiedene Pfade
 - Member = Klassifikationsknoten
 - Dadurch kompaktere Anfragen als mit SQL
 - SQL-artige Syntax
 - Wird von vielen kommerziellen Tools zur Kommunikation mit OLAP Datenbank benutzt

Multidimensional Expressions (MDX) - Struktur

- Dimensions (**SELECT**)
 - Angabe der darzustellenden Achsen der Ausgabe“tabellen“ **ON COLUMNS, ROWS, PAGES, CHAPTER, ...**
 - Achsenspezifikation muss eine Menge von Members beschreiben
Achsenbeschriftungen (geben auch Aggregationslevel vor)
 - Als explizit angegebene Menge (in {}) oder durch Funktionen
- Cube (**FROM**): Auswahl des Basis-Cubes für die Anfrage
- Slicer (**WHERE**)
 - Insb. Auswahl des betreffenden Fakten
 - Intuitiv zu lesen als „etwas zu tun haben mit“

```
SELECT <axis-spec1>, <axis-spec2>, ..  
FROM <cube-spec1>, <cube-spec2>, ...  
WHERE <slice-specification>
```

Multidimensional Expressions (MDX) - Beispiele

- Schachtelung zweier Dimensionen
- Bildet alle möglichen Kombinationen von Knotenwerten

```
SELECT CROSSJOIN( {Germany, France}
                  {Wein, Bier}) ON COLUMNS
      {Year.MEMBERS} on ROWS
FROM Sales
WHERE (Measures.Menge)
```

	Germany		France	
	Wein	Bier	Wein	Bier
1997				
1998				
...				

- Auswahl von Members über Bedingungen

```
SELECT FILTER(Germany.CHILDREN,
              ([2002], M.Menge) > ([2001], M.Menge)) ON COLUMNS
      Month.MEMBER ON ROWS
```

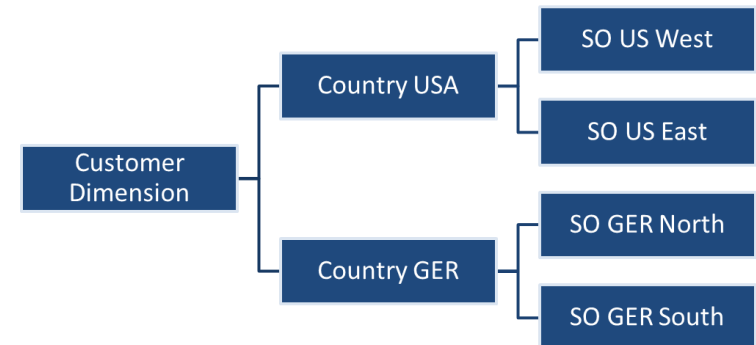
Slowly Changing Dimensions

In DWHs erfolgt die Identifikation von Kennzahlen durch die Ausprägungen der Dimensionen → **Annahme**: die Ausprägungen sind zeitinvariant

In der Realität erfolgen jedoch ...

- Änderungen von Attributausprägungen, Beziehungen und Entitäten im Zeitablauf
- Strukturveränderungen in den Dimensionen
- Schemaveränderungen in den Dimensionen

Herausforderung: Dokumentation der Änderungen über die Zeit und **Berücksichtigung der Änderung bei den Analysen** → **Berichtsformen**



Beispiel: Der Kunde Windy City Bikes in Chicago gehört bis zum 31.12.2010 zur SO US East

- Zum 01.01.2011 wird der Kunde der SO US West zugeordnet

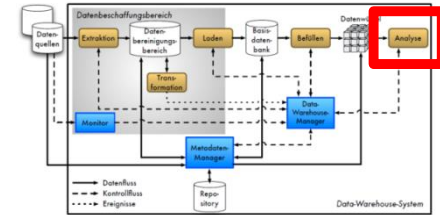
Customer	Location	Sales Organisation
Rocky Mountain Bikes	Denver	US West
Big Apple Bikes	New York City	US East
Philly Bikes	Philadelphia	US East
Peachtree Bikes	Atlanta	US East
Beantown Bikes	Boston	US East
Windy City Bikes	Chicago	US East
Furniture City Bikes	Grand Rapids	US East
Motown Bikes	Detroit	US East
SoCal Bikes	Irvine	US West
Silicon Valley Bikes	Palo Alto	US West
DC Bikes	Washington DC	US East
Northwest Bikes	Seattle	US West

Mapping 2010

Customer	Location	Sales Organisation
Rocky Mountain Bikes	Denver	US West
Big Apple Bikes	New York City	US East
Philly Bikes	Philadelphia	US East
Peachtree Bikes	Atlanta	US East
Beantown Bikes	Boston	US East
Windy City Bikes	Chicago	US West
Furniture City Bikes	Grand Rapids	US East
Motown Bikes	Detroit	US East
SoCal Bikes	Irvine	US West
Silicon Valley Bikes	Palo Alto	US West
DC Bikes	Washington DC	US East
Northwest Bikes	Seattle	US West

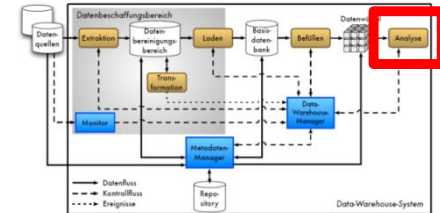
Mapping 2011

today



Berichtsformen: nach aktueller Struktur - as is

- Zusammenführung der Fakten mit den aktuellen Dimensionen
- Analyst: „Es kommt mir nur darauf an, zu welcher SO der Kunde **jetzt** gehört.“
- Auch die ursprünglich in einer anderen SO angefallenen Umsätze werden in der **neuen** SO angezeigt.
- Dies vereinfacht Vergleiche mit Vergangenheitszeiträumen (z.B. Vorjahr).



Current Structure

Mapping 2010

Customer	Location	Sales Organisation
...
Beantown Bikes	Boston	US East
Windy City Bikes	Chicago	US East
Furniture City Bikes	Grand Rapids	US East
...

Mapping 2011

Customer	Location	Sales Organisation
...
Beantown Bikes	Boston	US East
Windy City Bikes	Chicago	US West
Furniture City Bikes	Grand Rapids	US East
...

Revenue 2010 and 2011

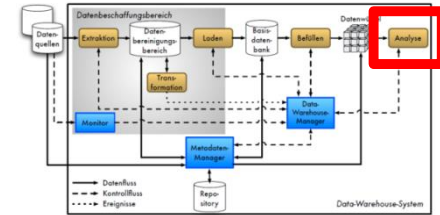
Customer	Calendar year	2010	2011
Location		\$	\$
...
Beantown Bikes	Boston	3.486.673	3.922.744
Windy City Bikes	Chicago	1.509.892	1.793.258
Furniture City Bikes	Grand Rapids	670.377	742.830
...

Current Structure

Sales Organisation	Customer	Calendar year	2010	2011
Location			\$	\$
US East
	Beantown Bikes	Boston	3.486.673	3.922.744
	Furniture City Bikes	Grand Rapids	670.377	742.830
US West	Windy City Bikes	Chicago	1.509.892	1.793.258
...

Berichtsformen: nach historischer Struktur - as posted

- Zusammenführung der Fakten mit den zum Zeitpunkt der Durchführung des Geschäftsvorfalls (e.g. Verkaufsvorgang) gültigen Dimensionen
- Analyst: „Es kommt mir nur darauf an, zu welcher SO der Kunde zum **Zeitpunkt des Verkaufsvorgangs** gehört hat.“
- Die alten Umsätze verbleiben bei der alten SO, ab dem Wechsel werden die Umsätze in der neuen SO angezeigt
- Szenario nennt man auch **Historische Wahrheit**



Customer	Location	Sales Organisation
...
Beantown Bikes	Boston	US East
Windy City Bikes	Chicago	US East
Furniture City Bikes	Grand Rapids	US East
...

Customer	Location	Sales Organisation
...
Beantown Bikes	Boston	US East
Windy City Bikes	Chicago	US West
Furniture City Bikes	Grand Rapids	US East
...

Revenue 2010 and 2011

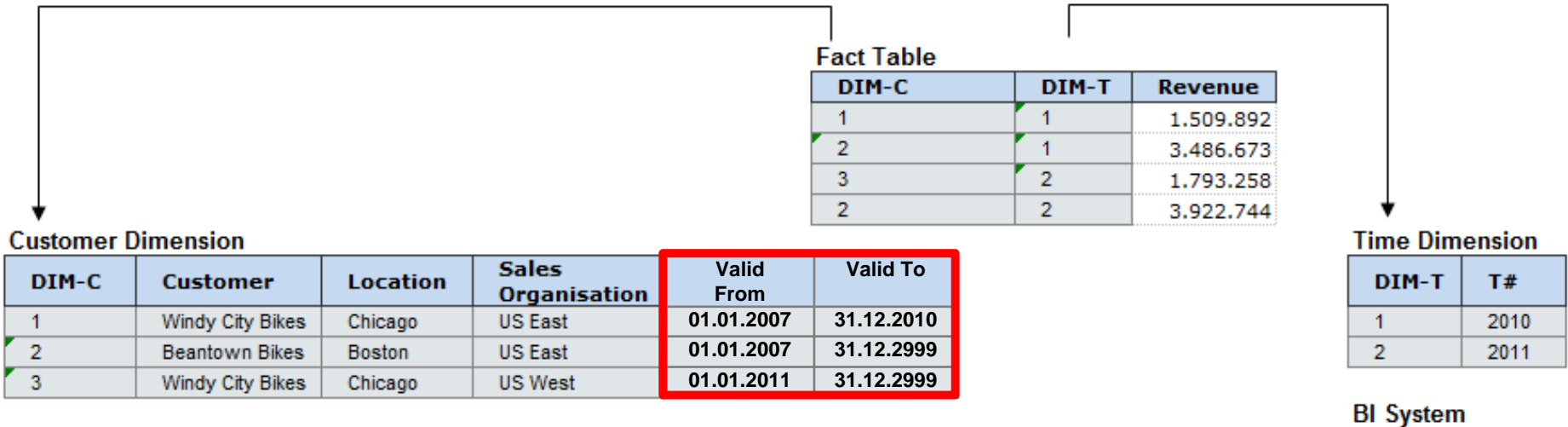
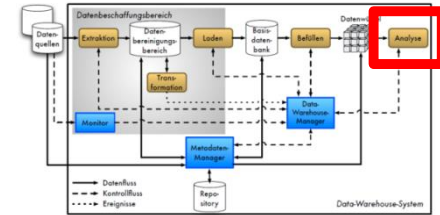
Customer	Calendar year	2010	2011
Location		\$	\$
...
Beantown Bikes	Boston	3.486.673	3.922.744
Windy City Bikes	Chicago	1.509.892	1.793.258
Furniture City Bikes	Grand Rapids	670.377	742.830
...

Transactional View

Sales Organisation	Customer	Calendar year	2010	2011
Location			\$	\$
US East
	Beantown Bikes	Boston	3.486.673	3.922.744
	Windy City Bikes	Chicago	1.509.892	
	Furniture City Bikes	Grand Rapids	670.377	742.830
US West	Windy City Bikes	Chicago		1.793.258
...

Berichtsformen: Realisierung über Zeitstempel

- Die technische Realisierung von Veränderungen in den Dimensionen/ Hierarchien erfolgt über das Einfügen eines Gültigkeitszeitraums für jeden Datensatz in die jeweilige Dimensionstabelle
- Hiermit können alle Berichtsformen realisiert werden



Source System

Customer	Location	Sales Organisation	T#	Revenue
Windy City Bikes	Chicago	US East	2010	1.509.892
Beantown Bikes	Boston	US East	2010	3.486.673
Windy City Bikes	Chicago	US West	2011	1.793.258
Beantown Bikes	Boston	US East	2011	3.922.744

(Köppen 2014)

Informationskonsument - Dashboard

- “A dashboard is a rich computer interface with charts, reports, visual indicators, and alert mechanisms that are consolidated into a dynamic and relevant information platform.” (Malik 2005)
- “Visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance.” (Few06)
- “multilayered application built on a business intelligence and data integration infrastructure that enables organizations to measure, monitor, and manage business performance more effectively”
→ **Corporate Performance Management** (Eckerson 2006)
- Ausgelegt für die direkte Nutzung am Bildschirm
- Komprimierung zentraler und relevanter Fakten auf eine oder wenige Bildschirmseiten
- Metapher/ Analogie zum Automobil-/ Flugzeugbau



Anforderungen an Dashboards: „SMART“

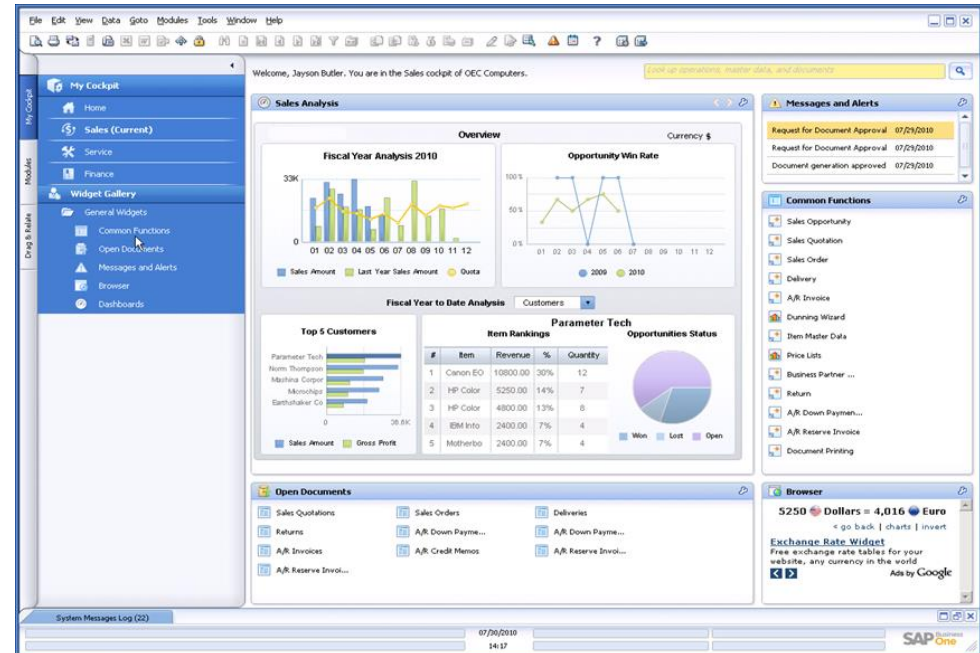
- **Synergetic:** Ein Dashboard muss ergonomisch und optisch effektiv gestaltet sein, um für den User relevante Informationen über die verschiedenen Aspekte in einer einzigen Bildschirmansicht darstellen zu können.
- **Monitor-KPIs:** Muss die kritischen KPIs anzeigen, die für eine effektive Entscheidungsfindung für den Einsatzbereich des Dashboards erforderlich sind.
- **Accurate:** Die dargestellten Informationen müssen sehr präzise sein, um das volle Vertrauen der Nutzer für das Dashboard gewinnen zu können. Die zu Grunde liegenden Dashboard-Daten müssen getestet und valide sein.
- **Responsive:** Vordefinierte Schwellenwerte müssen berücksichtigt und die visuelle Darstellung auf dem Dashboard (z. B. Sound-Alarme, E-Mails, Pager, Blinker) mit sofort nachvollziehbaren Meldungen ergänzt werden, die in kritischen Situationen die Aufmerksamkeit des Benutzers auf sich ziehen.
- **Timely:** Muss die neuesten Informationen für eine effektive Entscheidungsfindung anzeigen. Die Informationen müssen in Echtzeit und zur rechten Zeit angezeigt werden.

(Malik 2005)

Anforderungen an Dashboards – Darstellung

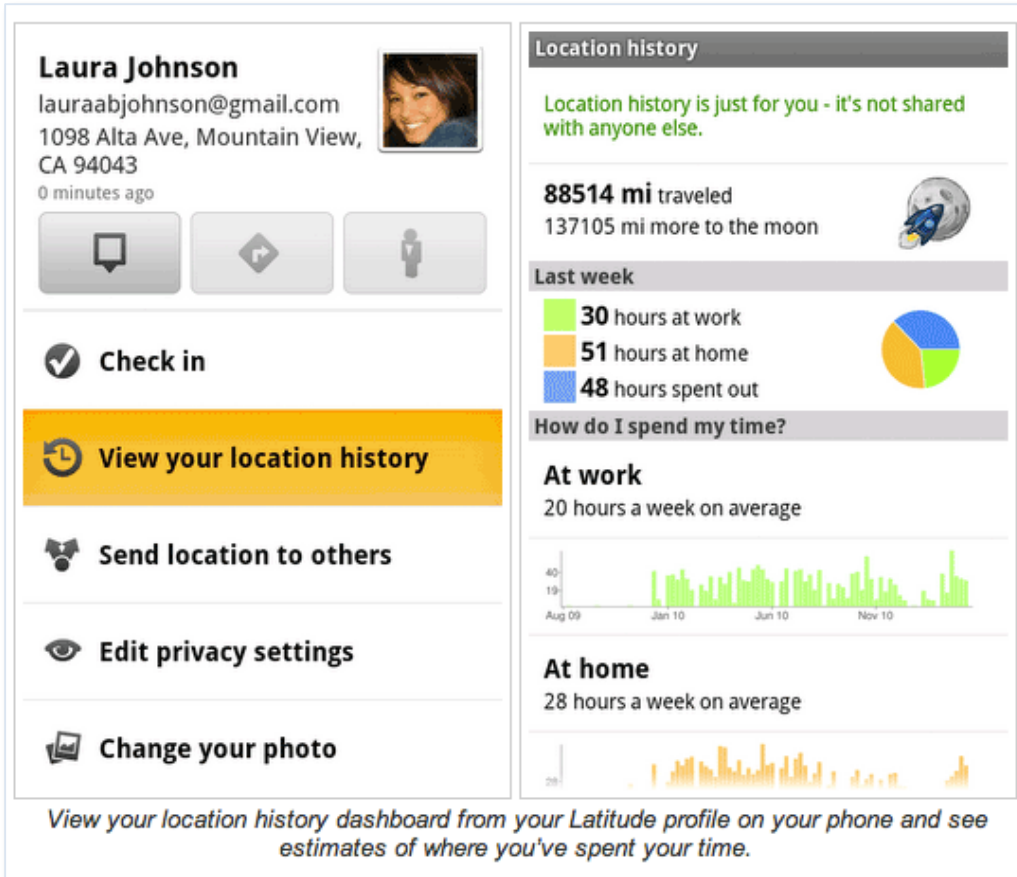
- **Ziel:** wesentliche Informationen verdichtet auf einen Blick wahrnehmbar zu präsentieren
- **Lösung:** Orientierung an der visuellen Wahrnehmung des Menschen bei der Aufnahme von Informationen → intuitiv verständliche/ anschauliche Gestaltungsarten
- **Eigenschaften:**
 - Komprimierte Darstellung
 - Konzentration auf wesentliche Informationen
 - Spezifische Lösung je nach Aufgabe u. Nutzertyp

Sales Dashboard

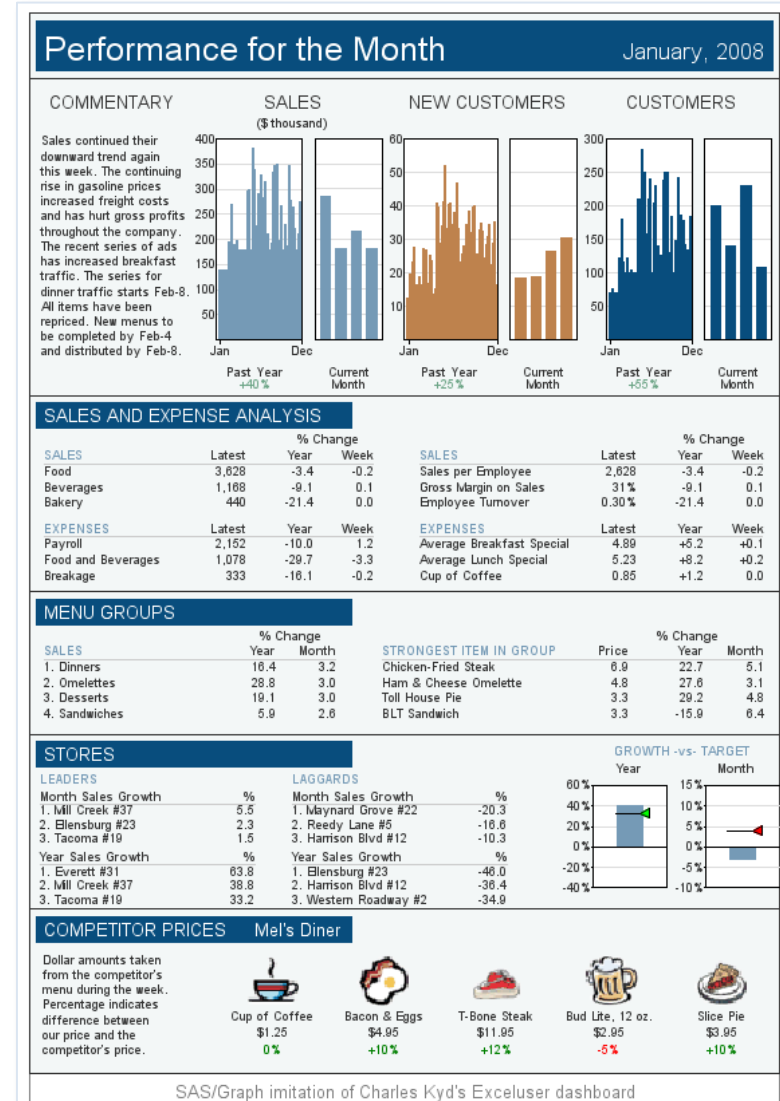


Dashboards: Weitere Beispiele

Google Location History Dashboard



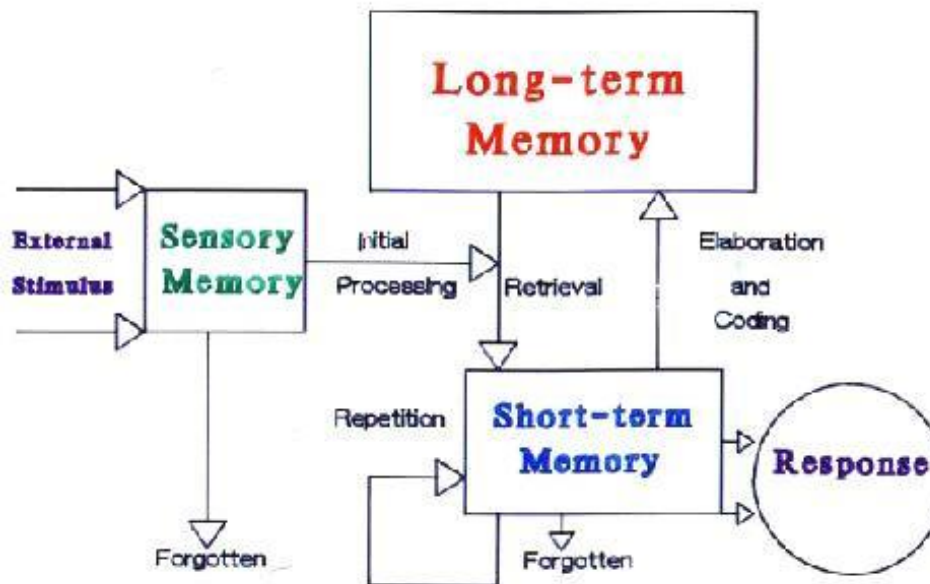
Restaurant Performance Dashboard



SAS/Graph imitation of Charles Kyd's Exceluser dashboard

Exkurs: Menschliche Wahrnehmung

- Das menschliche Gehirn kann mit einem Informationsstrom von 800 Zeichen pro Minute effektiv umgehen. Führungskräfte sind heute mit einer Informationsdichte von mindestens 4.000 Zeichen pro Minute konfrontiert. (Georges 1997)
- **Information Processing Theory:** short-time memory can only hold information for 15 to 30 seconds and only five to nine bits of information
Pictures take less space in our working memory than a written text
- **Cognitive Load Theory:** Over/ Underload of working memory during complex learning tasks



Aufgaben bei der Gestaltung von Dashboards

Aufgabe 1 - Anforderungsanalyse: Analyse des Einsatzbereichs/ Fachgebiets, insb. seines Vokabulars, seiner Begrifflichkeit und seiner Daten. Hier entsteht ein konzeptionelles Modell des Einsatzbereichs.

Aufgabe 2 – Sollkonzept/ Abstraktion:

- Überführung des Einsatzbereichs in ein abstraktes Modell des zukünftigen Dashboards
- Definition konkreter Vorgaben an die Visualisierung (Leistungen/Funktionen des Dashboards für den anvisierten Anwender sowie die Daten und Kennzahlen)

Aufgabe 3 – Visual Design/ Encoding: Definition der visuellen Darstellung und des Interaktionsmodells des Dashboards

- Auswahl der richtigen Darstellungsmittel für die Kennzahlen/ Daten
- Auswahl geeigneter Design-Elemente (z.B. Position, Größe, Farbe usw.)

Aufgabe 4 - Implementierung:

- Technische Umsetzung des Dashboards

1. Domain Problem Characterization

2. Data/ Operation Abstraction Design

3. Encoding/ Interaction Technique Design

4. Algorithm Design

(Munzner 2009)

Gestaltung von Dashboards: Aufgabe 1 - Anforderungsanalyse

Aufgabenanalyse

- **Ziel des Dashboards („Framing Statement“):** Das Framing-Statement definiert das Ziel des Dashboards. Es sollte an einem klaren Geschäftsziel ausgerichtet sein.
- **Operationen des Benutzers mit dem Dashboard/Basisfragen:** Die Operationen des Benutzers mit dem Dashboard definieren den Funktionsumfang des Dashboards
 - **Basis-Operationen:** retrieve value, filter, compute derived value, find extremum, sort, determine range, characterize distribution, find anomalies, cluster, correlate
 - **Praktischer Ansatz:** Formulierung von Fragen, auf die ein Anwender Antworten vom Dashboard erwartet. Die Fragen führen zu den Operationen des Dashboards

Informationsbedarfsanalyse

- Analyse und Auswahl der Daten/ Informationen auf denen das Dashboard aufsetzen soll
- **Ergebnisse:** Auswahl der kritischen Erfolgsfaktoren, Kennzahlen und Dimensionen, Zielwerte für Kennzahlen, Vergleichswerte für Kennzahlen, Schwellwerte und Alerts, Navigationsstruktur und Detaillierungsgrad

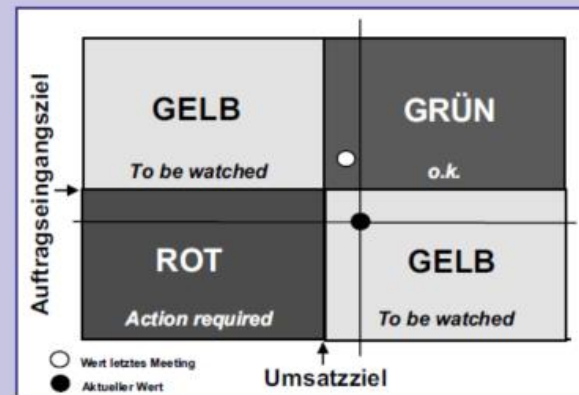
(Jacob 2011)

Gestaltung von Dashboards: Aufgabe 2 – Sollkonzept/ Abstraktion

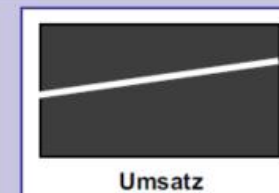
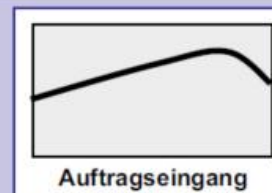
**Ebene 1: Gesamtstatus
des Logical View / zur Frage**

Erreichen wir unsere Ziele? (GELB)

**Ebene 2: Visualisierung
des zugrunde liegenden
Sachverhalts**



**Ebene 3: Visualisierung
der Details (Zeitreihe)**



**Ebene 4: Interaktive Online-
Analyse von Detailzahlen**



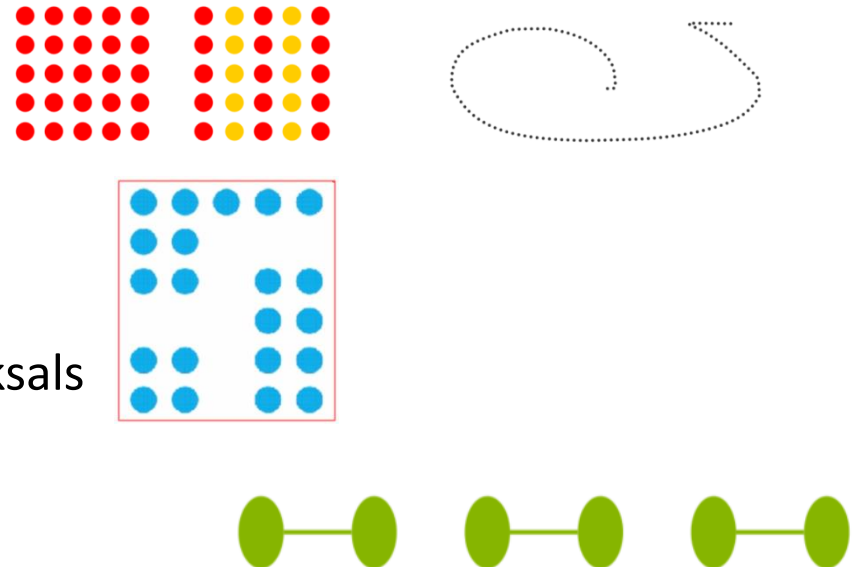
Daum 2006

Gestaltung von Dashboards: Aufgabe 3 – Visual Design/ Encoding

- Gestaltprinzipien, dienen dem strukturierten Anordnen von kleinen Teilen zu einem Ganzen stammen ursprünglich aus der Wahrnehmungspsychologie

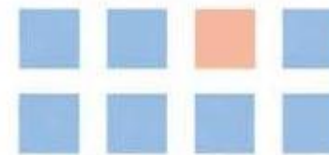
(Stapelkamp 2007)

- Das Prinzip der Ähnlichkeit
- Das Prinzip der Fortsetzung
- Das Prinzip der Nähe
- Das Prinzip des gemeinsamen Schicksals
- Das Prinzip der Vertrautheit
- Das Prinzip der Geschlossenheit
- Das Prinzip der gemeinsamen Region
- Das Prinzip der Verbundenheit der Elemente
- ...



Gestaltung von Dashboards: Visual Design/ Encoding - Aufmerksamkeit

- Für das Erkennen von Änderungen innerhalb von Elementen ist ein Erreichen der dazu nötigen **Aufmerksamkeit des Nutzers** essentiell
- d.h. die Änderungen (z.B. bei Systemausgaben) sollten so dargestellt werden, dass sie vom User direkt erkannt werden (Goldstein 2008)



Color hue



Position and alignment



Color brightness



Orientation



Color saturation



Size



Texture

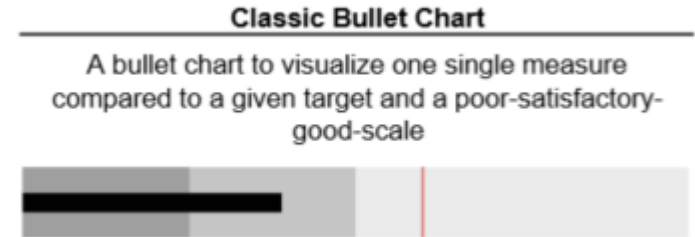


Shape

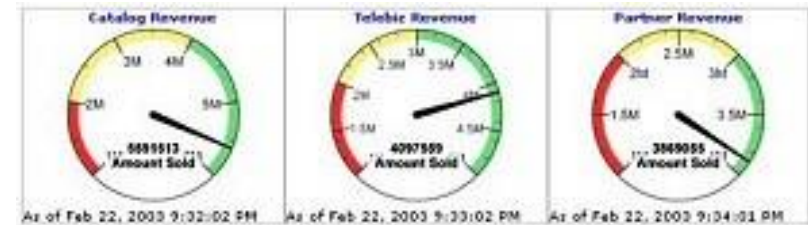
Gestaltung von Dashboards: Visual Encoding - Graphische Elemente

- Text, Diagramme (Linien, Kuchen, Säulen, Balken) oder

- Bullet Chart



- Tachometer



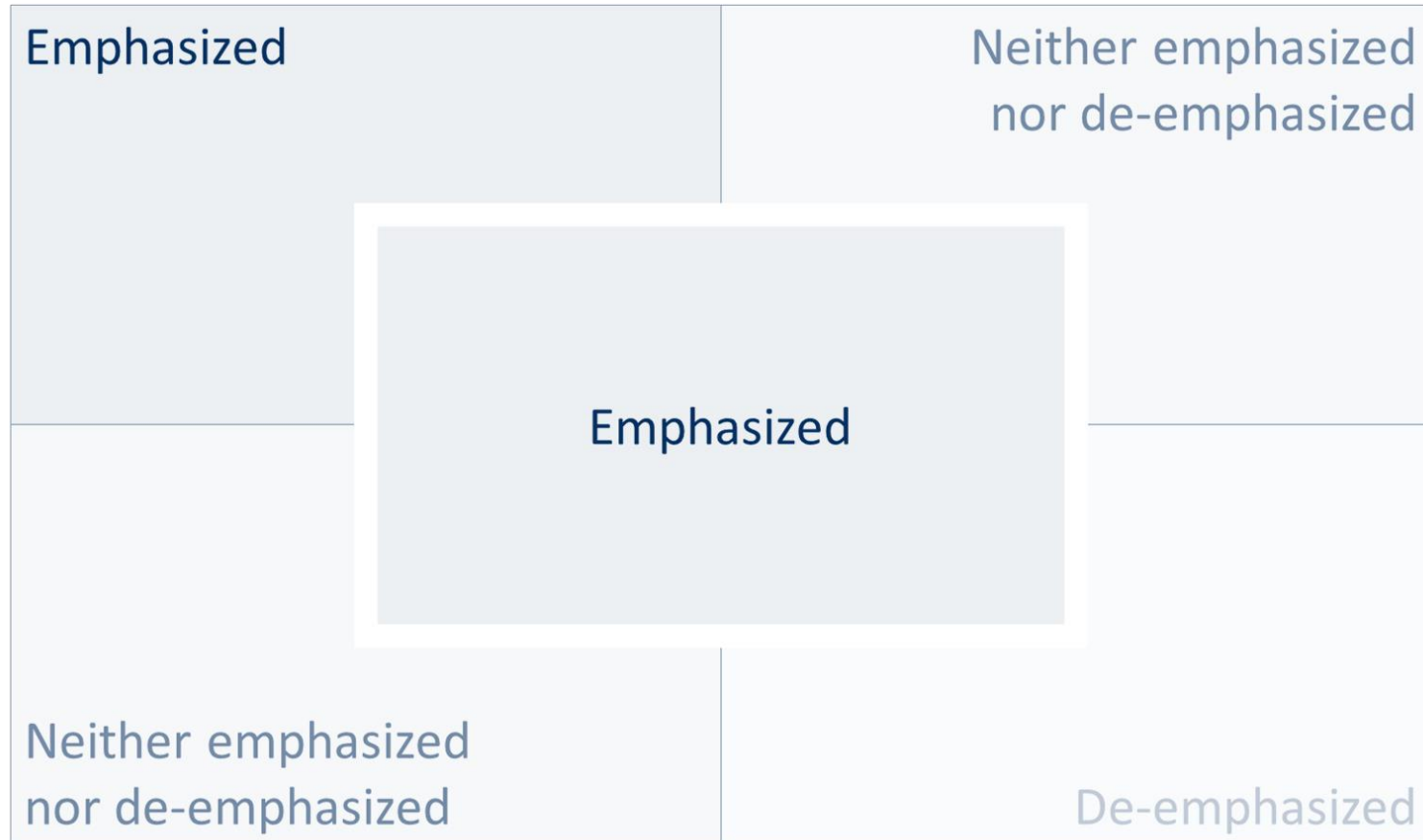
- Sparklines

- ...

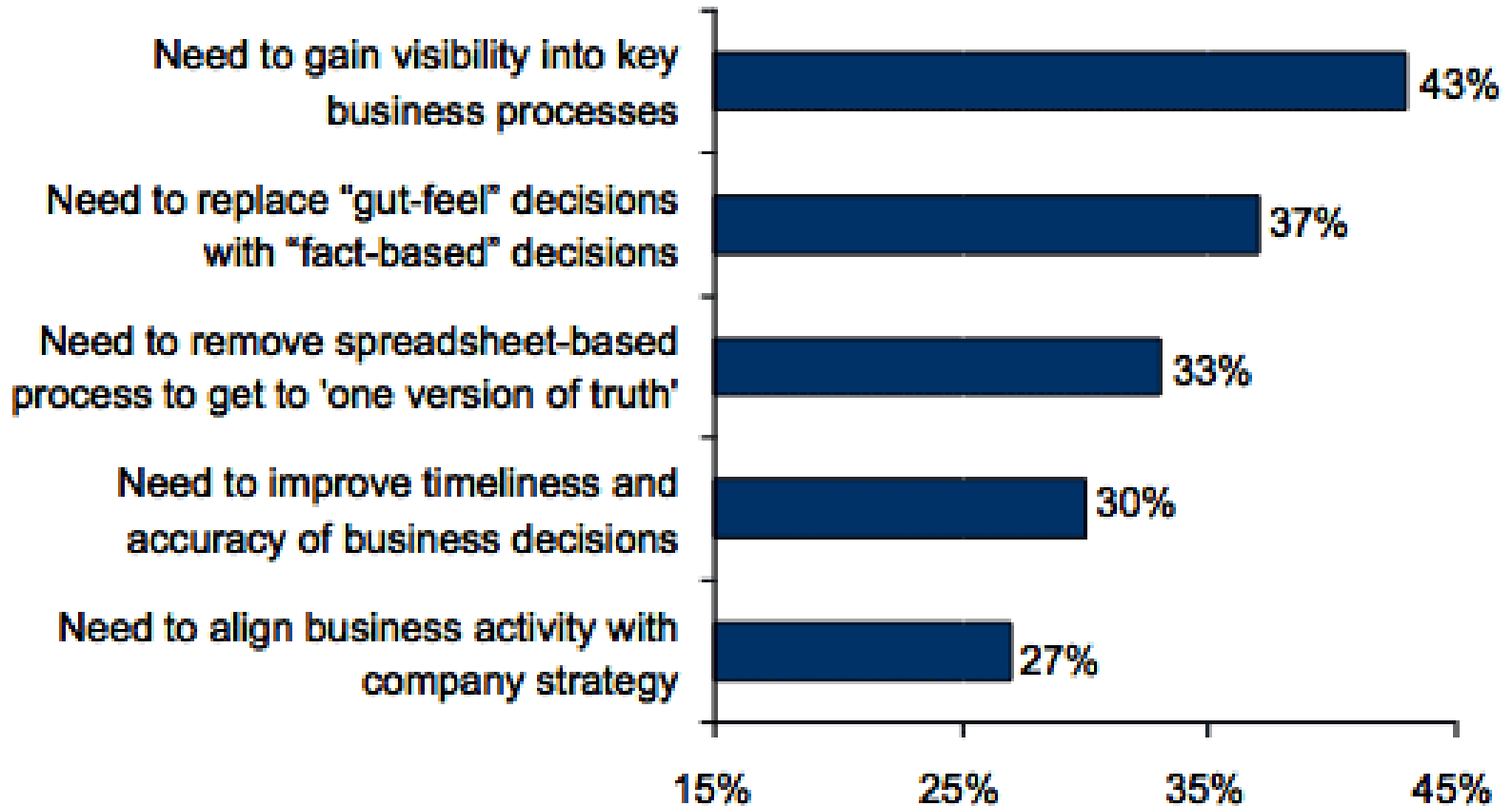
		Scenarios		...	
Measures	...	Actual	Plan	Actual-Budget Dev.	Actual-Budget Dev. %
Revenues		19.727.913	21.121.824	-1.393.911	-6,6%
Discounts		716.878	720.985	-4.108	-0,6%
Rebates		2.069.725	1.683.526	386.199	22,9%
Net Revenues		16.941.311	18.717.313	-1.776.002	-9,5%
Labour Cost		7.236.126	8.019.968	-783.842	-9,8%
Material Cost		351.778	314.312	37.466	11,9%
Margin		9.353.407	10.383.033	-1.029.626	-9,9%

Gestaltung von Dashboards: Visual Design/ Encoding - Screen Design

- 1) **Modi:** Kompetenzgrad der Gruppe (Anfänger, Fortgeschrittene, Experte)
- 2) **Orientierung**
- 3) **Positionierung**



Treiber für die Implementierung von Dashboards



Percentage of Respondents, n = 285

Source: Aberdeen Group, May 2009