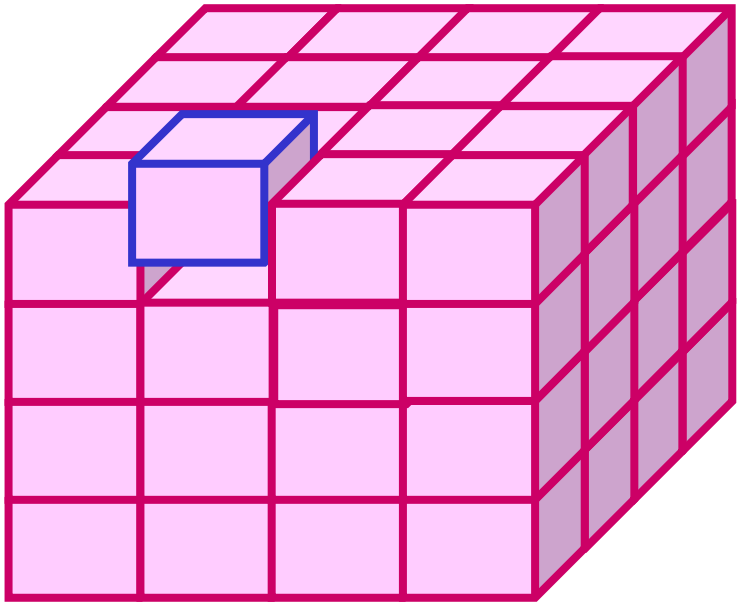
Data Warehousing

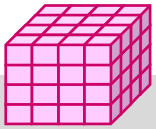
Kapitel 2: Architektur von DWH-Systemen

Michael Hartung
Sommersemester 2011

Universität Leipzig
Institut für Informatik

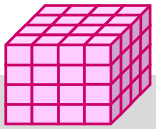
http://dbs.uni-leipzig.de



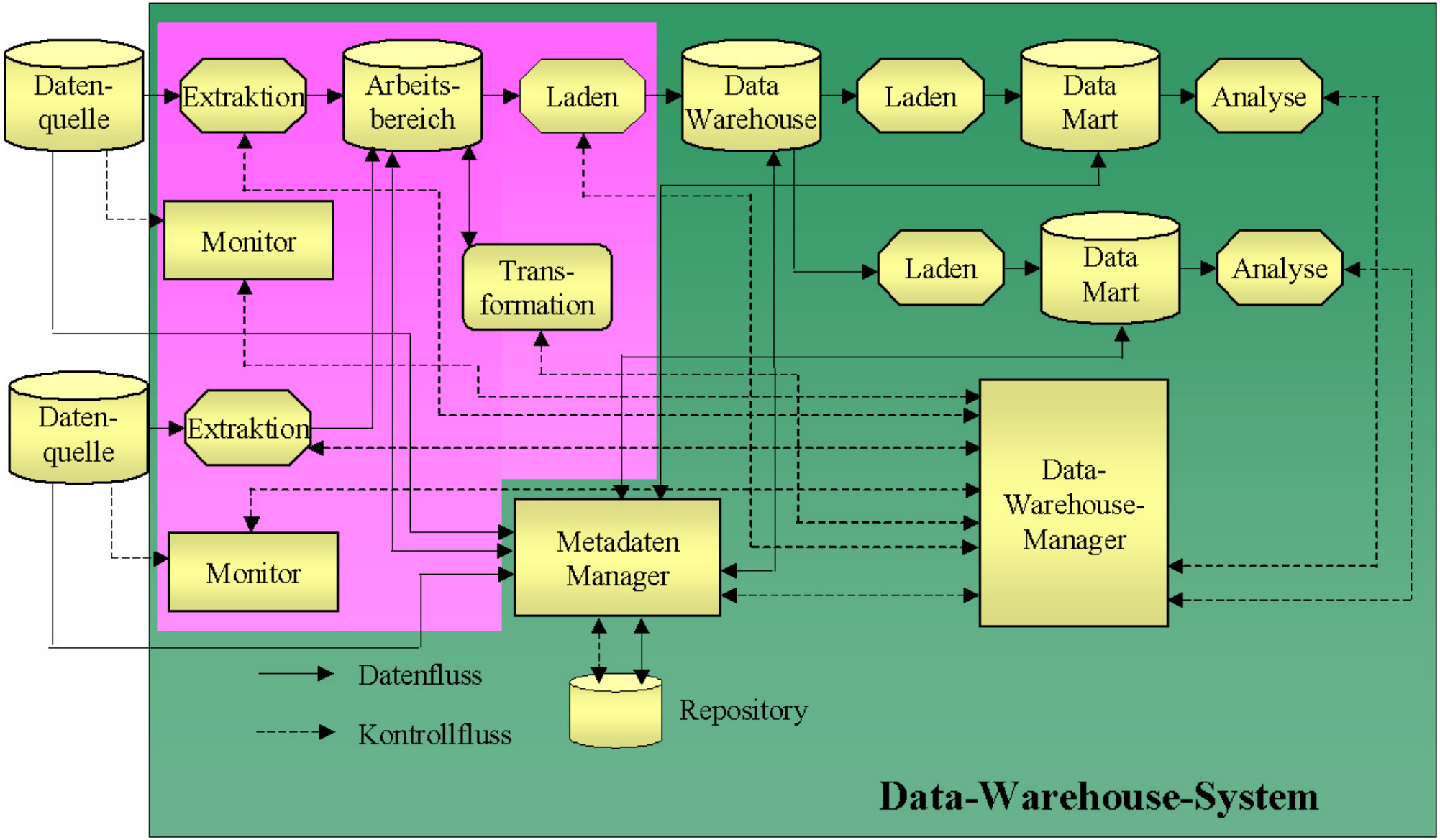


2. Architektur von Data Warehouse-Systemen

- Referenzarchitektur
  - Scheduler
  - Datenquellen
  - Datenextraktion
  - Transformation und Laden
- Abhängige vs. unabhängige Data Marts
- Metadatenverwaltung
  - Klassifikation von Metadaten (technische vs. fachliche Metadaten)
  - CWM: Common Warehouse Model
  - Interoperabilitätsmechanismen
- Operational Data Store (ODS)
- Master Data Management (MDM)
- Column Stores



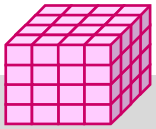
DW-Referenzarchitektur



Quelle: Bauer/Günzel, 2004

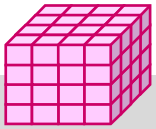
Phasen des Data Warehousing

1. Überwachung der Quellen auf Änderungen durch Monitore
2. Kopieren der relevanten Daten mittels Extraktion in temporären Arbeitsbereich
3. Transformation der Daten im Arbeitsbereich (Bereinigung, Integration)
4. Kopieren der Daten ins Data Warehouse (DW) als Grundlage für verschiedene Analysen
5. Laden der Daten in Data Marts (DM)
6. Analyse: Operationen auf Daten des DW oder DM



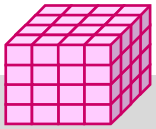
Datenquellen

- Lieferanten der Daten für das Data Warehouse (gehören nicht direkt zum DW)
- Merkmale
  - intern (Unternehmen) oder extern (z.B. Internet)
  - ggf. kostenpflichtig
  - i.a. autonom
  - i.a. heterogen bzgl. Struktur, Inhalt und Schnittstellen (Datenbanken, Dateien)
- Qualitätsforderungen:
  - Verfügbarkeit von Metadaten
  - Konsistenz (Widerspruchsfreiheit)
  - Korrektheit (Übereinstimmung mit Realität)
  - Vollständigkeit (z.B. keine fehlenden Werte oder Attribute)
  - Aktualität
  - Verständlichkeit
  - Verwendbarkeit
  - Relevanz



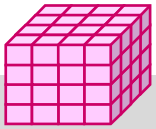
Data-Warehouse-Manager/Scheduler

- Ablaufsteuerung: Initiierung, Steuerung und Überwachung der einzelnen Prozesse
- Initiierung des Datenbeschaffungsprozesses und Übertragung der Daten in Arbeitsbereich
  - in regelmäßigen Zeitabständen (jede Nacht, am Wochenende etc.)
  - bei Änderung einer Quelle: Start der entsprechenden Extraktionskomponente
  - auf explizites Verlangen durch Administrator
- Fehlerfall: Dokumentation von Fehlern, Wiederanlaufmechanismen
- Zugriff auf Metadaten aus dem Repository
  - Steuerung des Ablaufs
  - Parameter der Komponenten



Datenextraktion

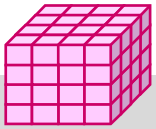
- Monitore: Entdeckung von Datenmanipulationen in einer Datenquelle
  - interne Datenquellen: aktive Mechanismen
  - externe Datenquellen: Polling / periodische Abfragen
- Extraktionskomponenten: Übertragung von Daten aus Quellen in Arbeitsbereich
  - periodisch
  - auf Anfrage
  - ereignisgesteuert (z.B. bei Erreichen einer definierten Anzahl von Änderungen)
  - sofortige Extraktion
- unterschiedliche Funktionalität der Quellsysteme
- Nutzung von Standardschnittstellen (z.B. ODBC) oder Eigenentwicklung
- Performance-Probleme bei großen Datenmengen
- Autonomie der Quellsysteme ist zu wahren



Datenextraktion: Strategien

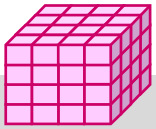
- Snapshots: periodisches Kopieren des Datenbestandes in Datei
- Trigger
  - Auslösen von Triggern bei Datenänderungen und Kopieren der geänderten Tupel
- Log-basiert
  - Analyse von Transaktions-Log-Dateien der DBMS zur Erkennung von Änderungen
- Nutzung von DBMS-Replikationsmechanismen

	Autonomie	Performanz	Nutzbarkeit
Snapshot			
Log			
Trigger			
Replikation			



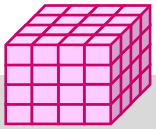
Datentransformation und Laden

- *Arbeitsbereich* (engl.: *Staging Area*)
  - Temporärer Zwischenspeicher zur Integration und Bereinigung
  - Laden der Daten ins DW erst nach erfolgreichem Abschluss der Transformation
  - Keine Beeinflussung der Quellen oder des DW
  - Keine Weitergabe fehlerbehafteter Daten
- *Transformationskomponente*: Vorbereitung der Daten für Laden
  - Vereinheitlich von Datentypen, Datumsangaben, Maßeinheiten, Kodierungen etc.
  - Data Cleaning und Scrubbing: Beseitigung von Verunreinigungen, fehlerhafte oder fehlende Werte, Redundanzen, veralteten Werte
  - Data Auditing:Anwendung von Data-Mining-Verfahren zum Aufdecken von Regeln und Aufspüren von Abweichungen
- *Ladekomponente*: Übertragung der bereinigten und aufbereiteten (z.B. aggregierten) Daten in DW
  - Nutzung spezieller Ladewerkzeuge (z.B. Bulk Loader)
  - Historisierung: zusätzliches Abspeichern geänderter Daten anstatt Überschreiben
  - Offline vs. Online-Laden (Verfügbarkeit des DW während des Ladens)

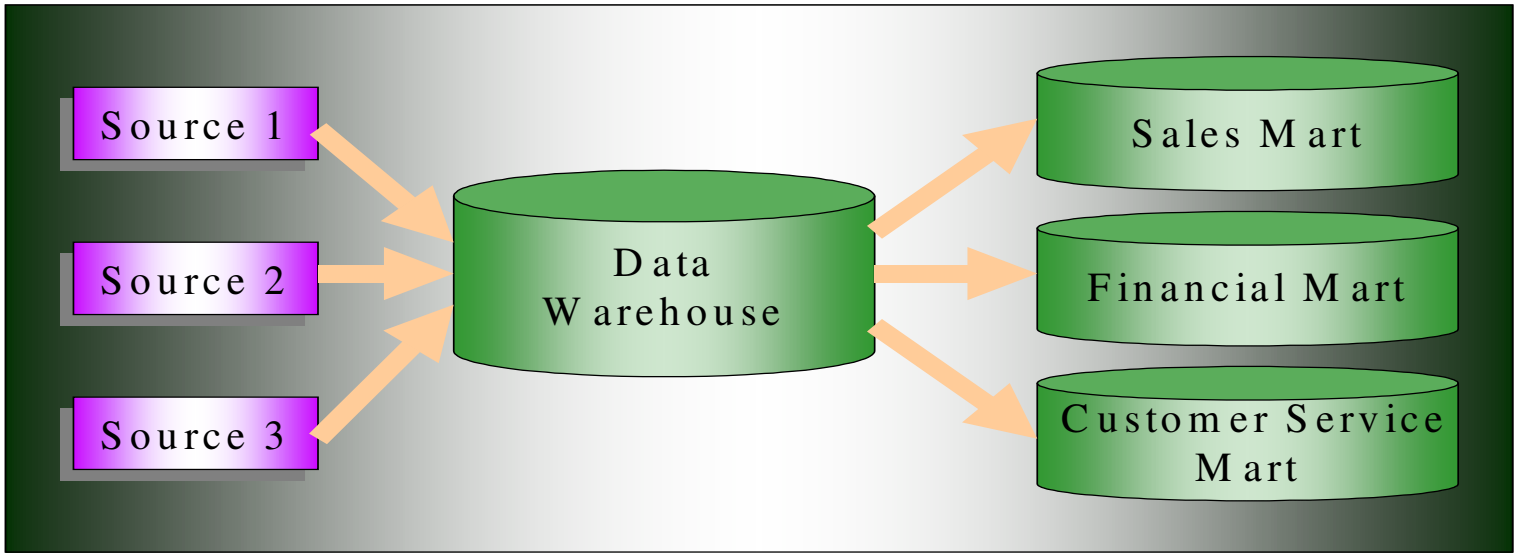


Data Marts

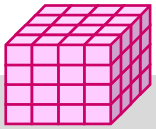
- Was ist eine Data Mart?
  - eine Teilmenge des Data Warehouse
  - inhaltliche Beschränkung auf bestimmten Themenkomplex oder Geschäftsbereich
- führt zu verteilter DW-Lösung
- Gründe für Data Marts
  - Performance: schnellere Anfragen, weniger Benutzer, Lastverteilung
  - Eigenständigkeit, Datenschutz
  - ggf. schnellere Realisierung
- Probleme
  - zusätzliche Redundanz
  - zusätzlicher Transformationsaufwand
  - erhöhte Konsistenzprobleme
- Varianten
  - Abhängige Data Marts
  - Unabhängige Data Marts



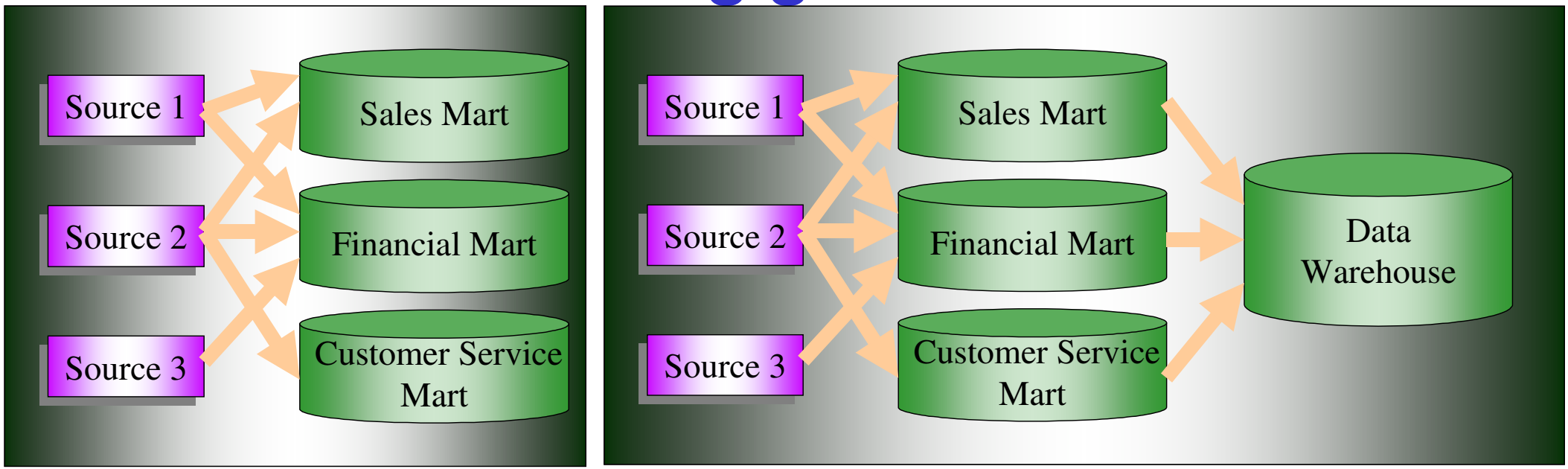
Abhängige Data Marts



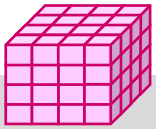
- „Nabe- und Speiche“-Architektur (hub and spoke)
- Data Marts sind Extrakte aus dem zentralen Warehouse
  - strukturelle Ausschnitte (Teilschema, z.B. nur bestimmte Kennzahlen)
  - inhaltliche Extrakte (z.B. nur bestimmter Zeitraum, bestimmte Filialen ...)
  - Aggregierung (geringere Granularität), z.B. nur Monatssummen
- Vorteile:
  - relativ einfach ableitbar (Replikationsmechanismen des Warehouse-DBS)
  - Analysen auf Data Marts sind konsistent mit Analysen auf Warehouse
- Nachteil: Entwicklungsdauer (Unternehmens-DW zunächst zu erstellen)



Unabhängige Data Marts

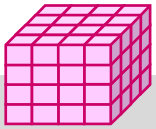


- Variante 1: kein zentrales, unternehmensweites DW
  - wesentlich einfachere und schnellere Erstellung der DM verglichen mit DW
  - Datenduplizierung zwischen Data Marts, Gefahr von Konsistenzproblemen
  - Aufwand wächst proportional zur Anzahl der DM
  - schwierigere Erweiterbarkeit
  - keine unternehmensweite Analysemöglichkeit
- Variante 2: unabhängige DM + Ableitung eines DW aus DM
- Variante 3: unabhängige DM + Verwendung gemeinsamer Dimensionen



Metadaten-Verwaltung

- Anforderungen an Metadaten-Verwaltung / Repository
  - vollständige Bereitstellung aller relevanten Metadaten auf aktuellem Stand
  - flexible Zugriffsmöglichkeiten (DB-basiert) über mächtige Schnittstellen
  - Versions- und Konfigurationsverwaltung
  - Unterstützung für technische und fachliche Aufgaben und Nutzer
  - aktive Nutzung für DW-Prozesse (Datentransformation, Analyse)
- Realisierungsformen
  - werkzeugspezifisch: fester Teil von Werkzeugen
  - allgemein einsetzbar: generisches und erweiterbares Repository-Schema (Metadaten-Modell)
- zahlreiche proprietäre Metadaten-Modelle
- Standardisierungsbemühungen
  - Open Information Model (OIM): Metadata Coalition (MDC) - wurde 2000 eingestellt
  - Common Warehouse Metamodel (CWM): Object Management Group (OMG)
- häufig Integration von bzw. Austausch zwischen dezentralen Metadaten-Verwaltungssystemen notwendig



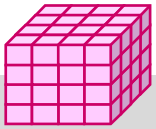
Metadaten: Beispiel

Personal : Tabelle

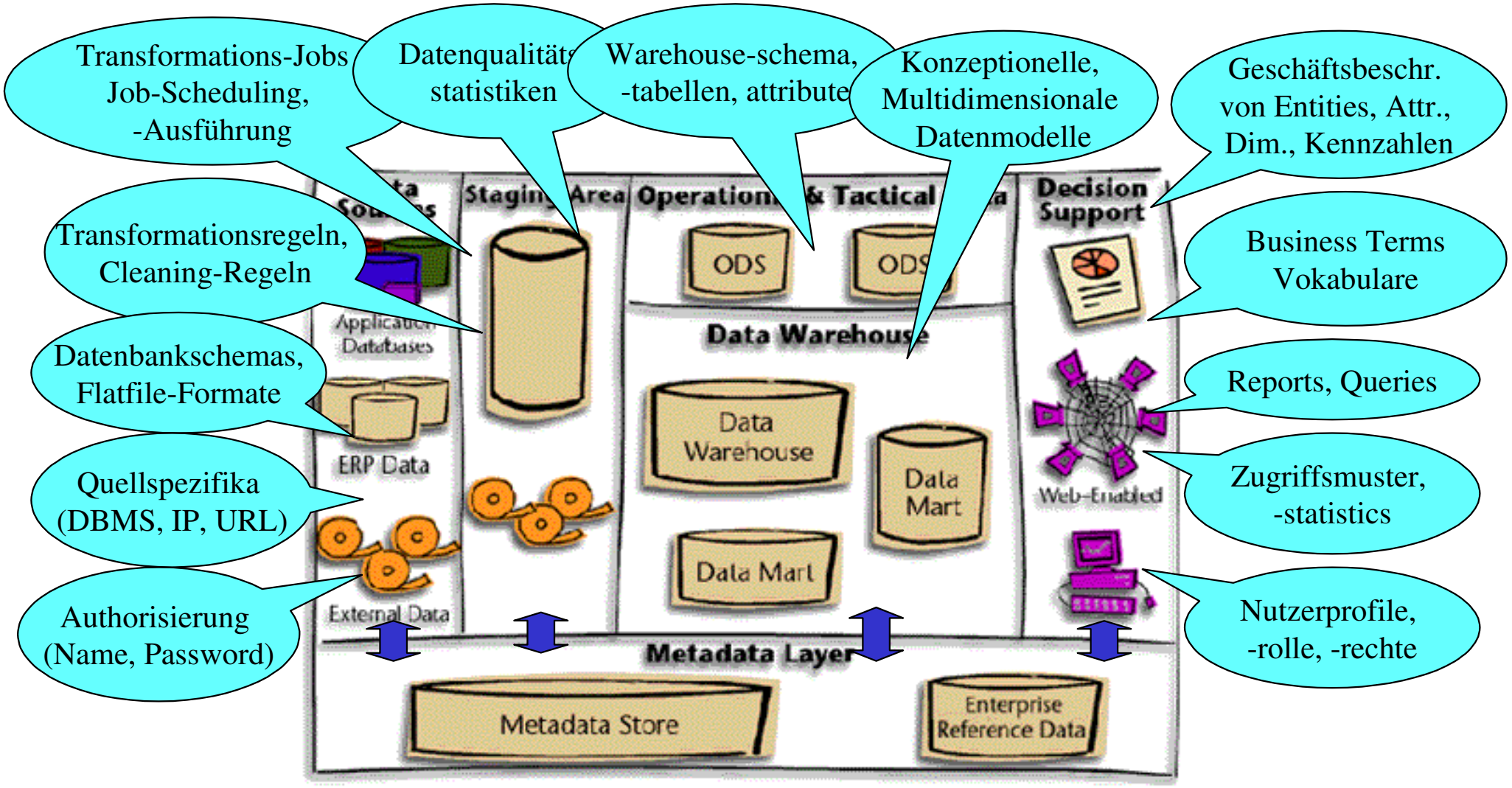
	Feldname	Felddatentyp	Beschreibung
🔑	Personal-Nr	AutoWert	Nummer, die einem neuen Angestellten automatisch zugewiesen wird.
	Nachname	Text	
	Vorname	Text	
	Position	Text	Position des Angestellten.
	Anrede	Text	In Begrüßungen verwendete Anrede.
	Geburtsdatum	Datum/Uhrzeit	
	Einstellung	Datum/Uhrzeit	
	Straße	Text	Straße oder Postfach.
	Ort	Text	
	Region	Text	Bundesland oder Provinz.
	PLZ	Text	
	Land	Text	
	Telefon privat	Text	Telefonnummer mit (internationaler) Vorwahl.
	Durchwahl Büro	Text	Interne Durchwahlnummer zum Büro
	Foto	O	
	Bemerkungen	M	
	Vorgesetzte(r)	Z	

Personal : Tabelle

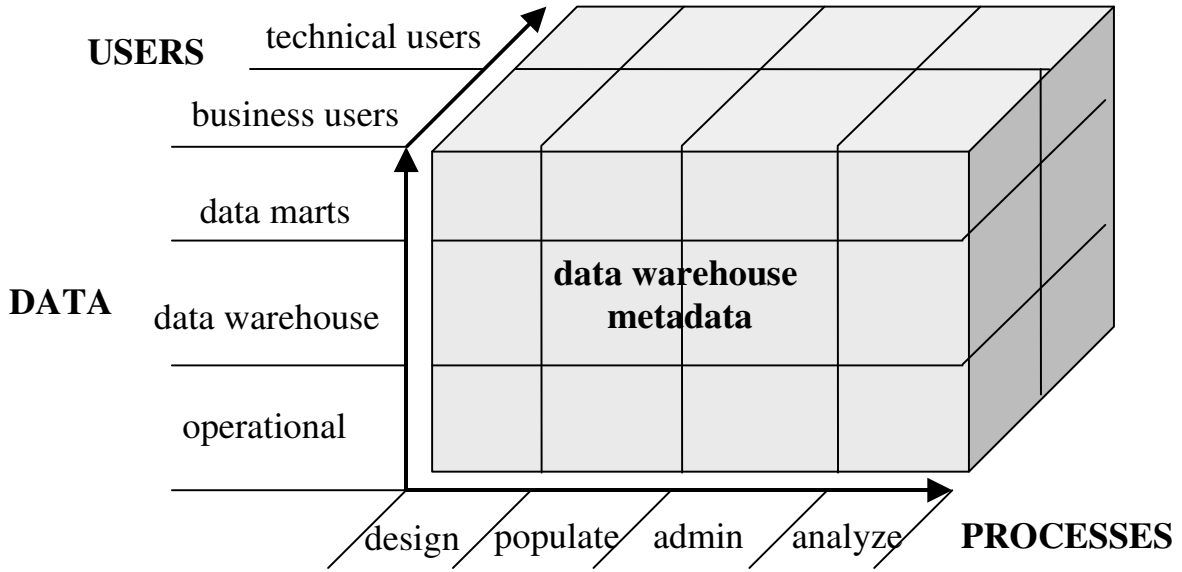
	Personal-Nr	Nachname	Vorname	Position	Anrede	Geburtsdatum
▶	1	Davolio	Nancy	Vertriebsmitarbeiterin	Frau	08. Dez. 48
	2	Fuller	Andrew	Geschäftsführer	Herr	19. Feb. 52
	3	Leverling	Janet	Vertriebsmitarbeiterin	Frau	30. Aug. 63
	4	Peacock	Margaret	Vertriebsmitarbeiterin	Frau	19. Sep. 37
	5	Buchanan	Steven	Vertriebsmanager	Herr	04. Mrz. 55
	6	Suyama	Michael	Vertriebsmitarbeiter	Herr	02. Jul. 63
	7	King	Robert	Vertriebsmitarbeiter	Dr.	29. Mai. 60
	8	Callahan	Laura	Vertriebskoordinatorin	Frau	09. Jan. 58
	9	Dodsworth	Anne	Vertriebsmitarbeiterin	Frau	27. Jan. 66

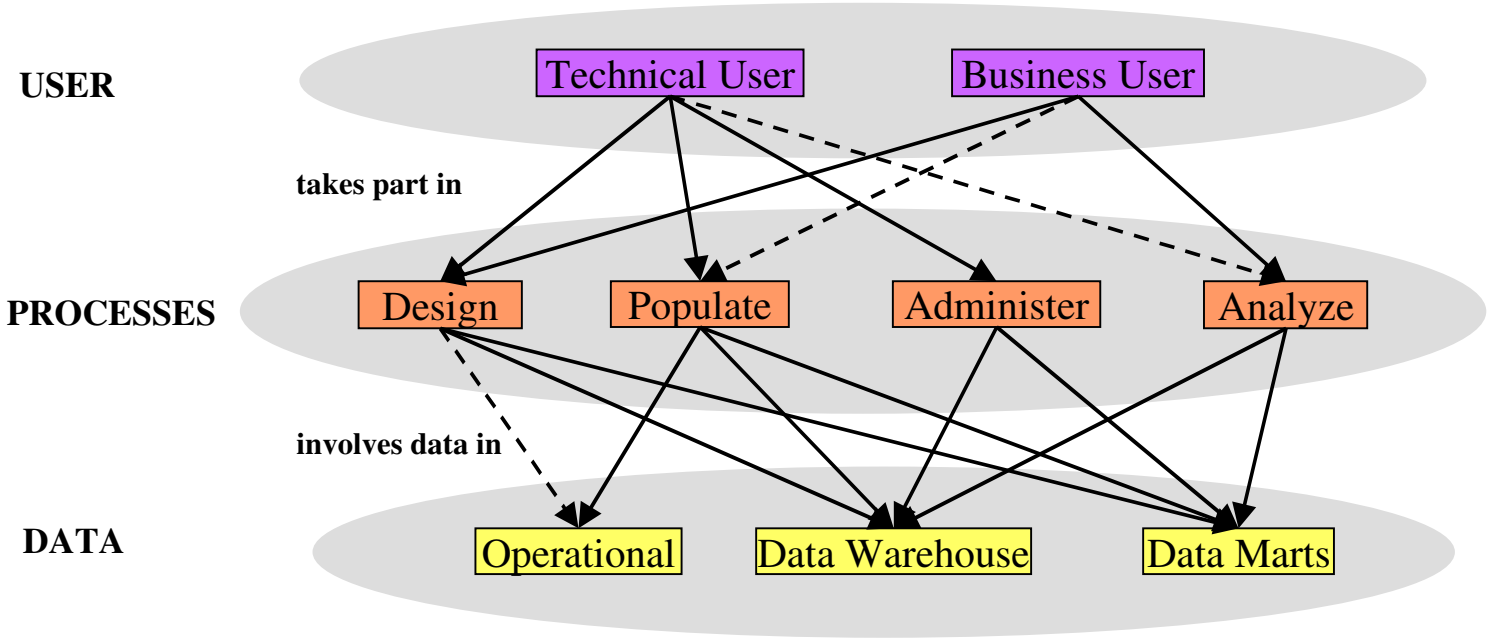


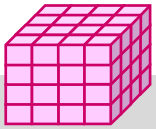
Metadaten im Data Warehouse-Kontext



Klassifikation von DW-Metadaten

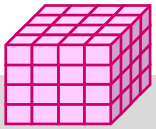






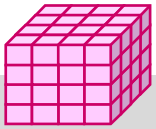
Technische Metadaten

- Schemata
  - Datenbank-Schemata, Dateiformate
- Quell-, Ziel-Systeme
  - technische Charakteristika für Zugriff (IP, Protokoll, Benutzername und Passwort, etc.)
- Datenabhängigkeiten: (technische) Mappings
  - Operationale Systeme <-> Data Warehouse, Data Marts: Datentransformations-Regeln
  - Data Warehouse, Data Marts <-> Datenzugriff-Tools: Technische Beschreibung von Queries, Reports, Cubes (SQL, Aggregation, Filters, etc.)
- Warehouse-Administration (Datenaktualisierung, -archivierung, Optimierung)
  - Systemstatistiken (Usage Patterns, nutzer-/gruppenspezifische CPU-/ IO-Nutzung, ...) für Resourcenplanung und Optimierung
  - Häufigkeit (Scheduling), Logging-Information, Job-Ausführungsstatus
  - Regeln, Funktion für Datenselektion für Archivierung



Business-Metadaten

- Informationsmodelle, konzeptuelle Datenmodelle
- Unternehmens-/Branchen-spezifische Business Terms, Vokabulare, Terminologien
- Abbildungen zwischen Business Terms und Warehouse/Data Mart-Elementen (Dimensionen, Attribute, Fakten)
- Geschäftsbeschreibung von Queries, Reports, Cubes, Kennzahlen
- Datenqualität
  - Herkunft (*lineage*): aus welchen Quellen stammen die Daten? Besitzer?
  - Richtigkeit (*accuracy*): welche Transformation wurden angewendet?
  - Aktualität (*timeliness*): wann war der letzte Aktualisierungsvorgang?
- Personalisierung
  - Beziehungen zw. Nutzer, Nutzerrollen, Informationsobjekten, Interessengebieten und Aktivitäten
  - Zuordnung von Nutzer zu Rollen, von Rollen zu Aktivitäten bzw. zu Interessengebieten, und von Aktivitäten zu Informationsobjekten und Interessengebieten



Business Metadaten: Beispiel

- Business Terms für Versicherungsindustrie

Liability Insurance:

Insurance covering the legal liability of the insured resulting from injuries to a third party to their body or damage to their property.

Life Insurance:

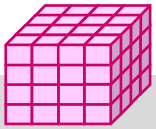
Insurance providing payment of a specified amount on the insured's death, either to his or her estate or to a designated beneficiary.

Liquor Liability Insurance:

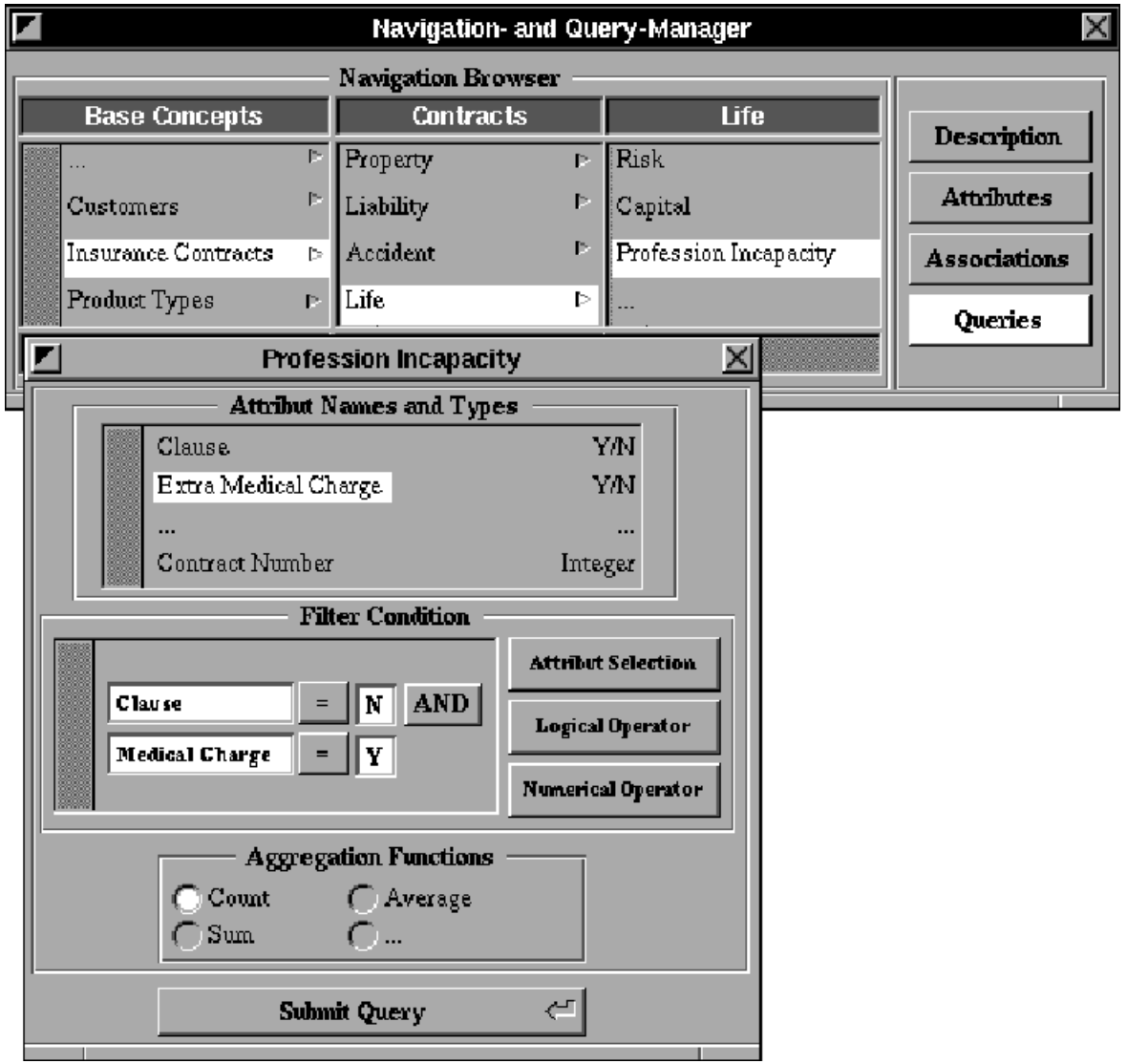
Provides protection for the owners of an establishment that sells alcoholic beverages against liability arising out of accidents caused by intoxicated customers.

Long-Term Disability Insurance:

Insurance to provide a reasonable replacement of a portion of an employee's earned income lost through serious illness or injury during the normal work career:

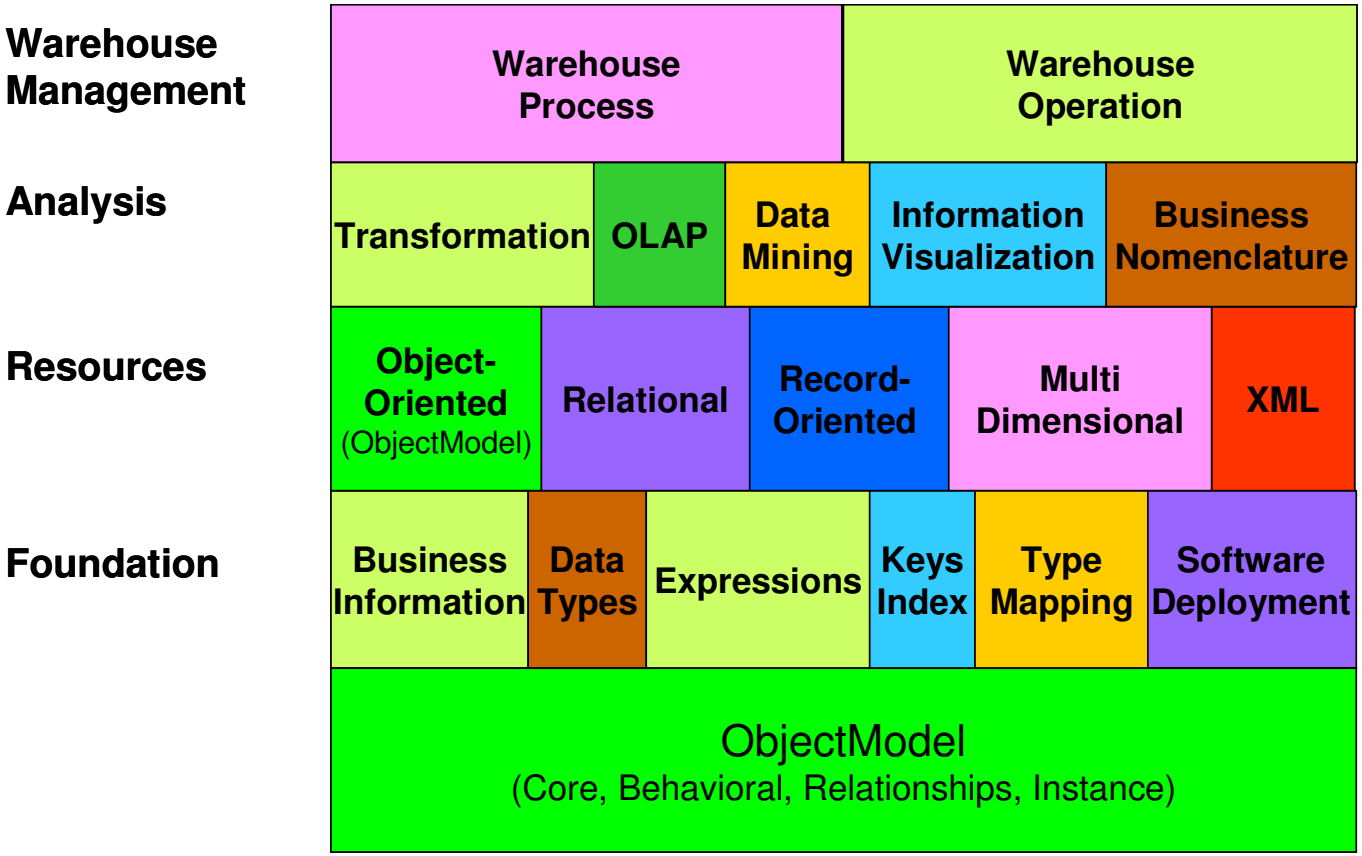


Business Metadaten: Beispiel

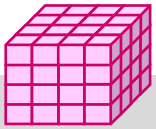


Common Warehouse Metamodel (CWM)

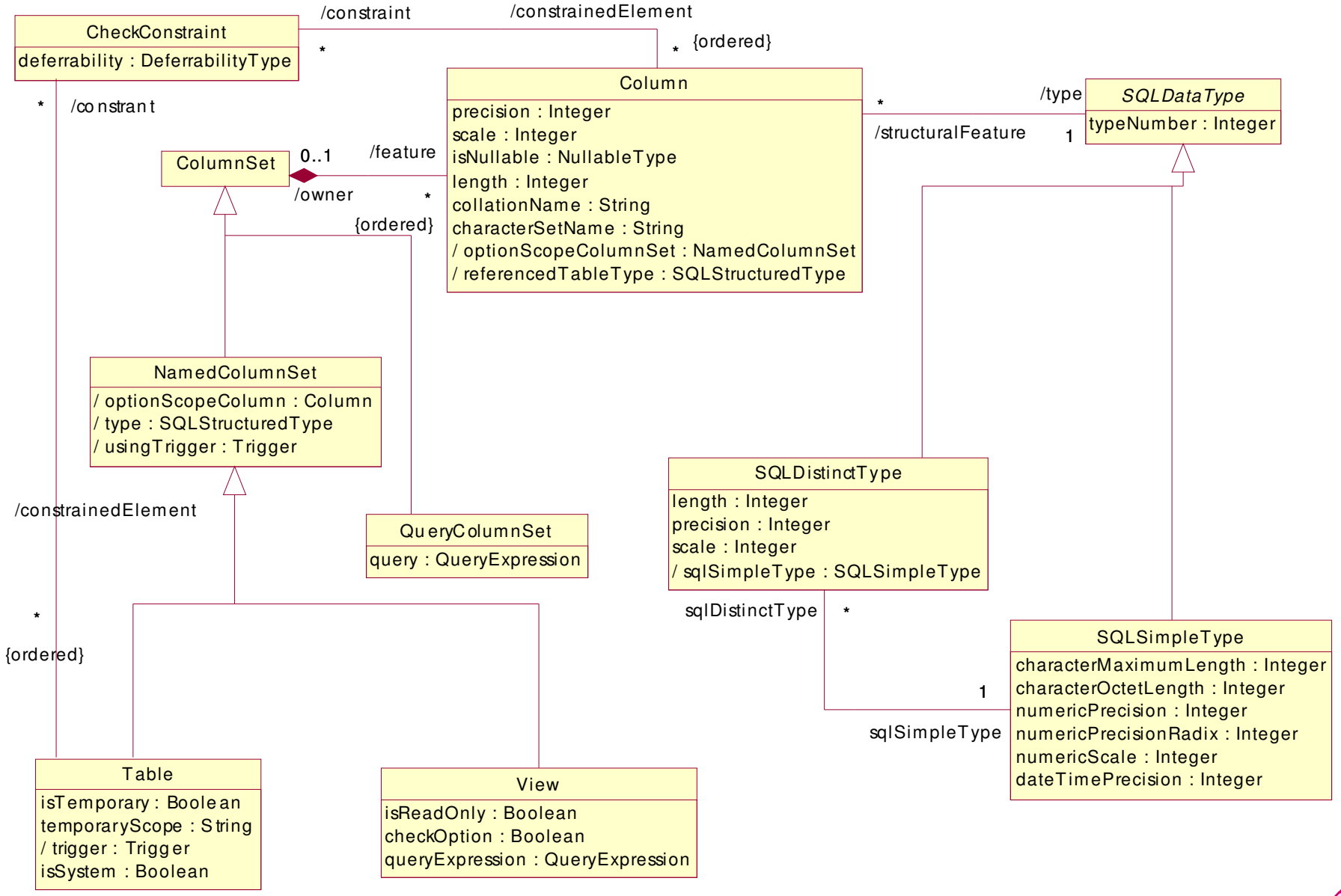
- umfassende UML-basierte Metadaten-Modelle für Data Warehousing
- OMG-Standard
  - CWM 1.0: 2001
  - CWM 1.1: 2002

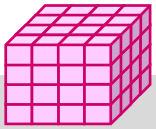


- Web-Infos: *www.omg.org/cwm* , *www.cwmforum.org*
- geringe Produktunterstützung

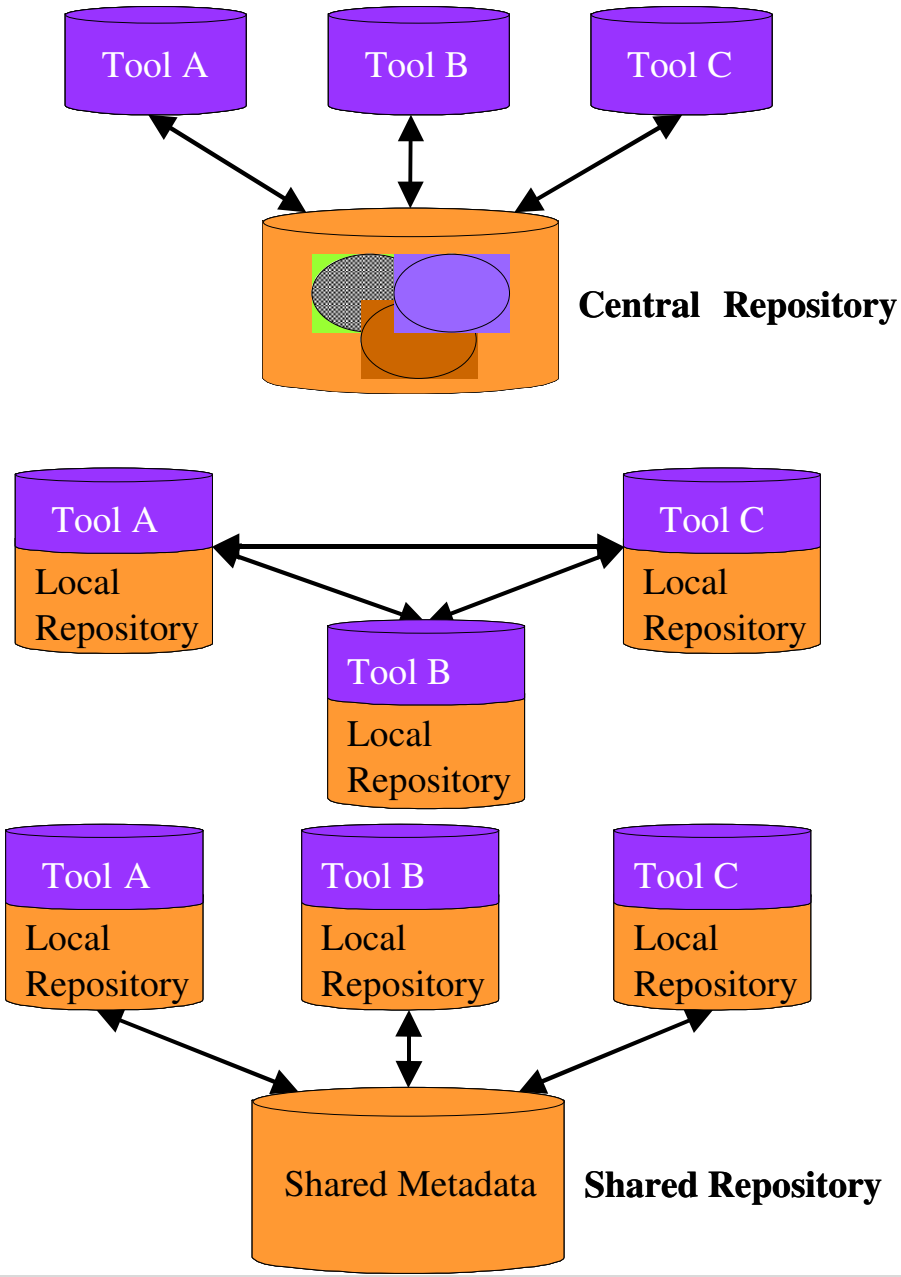


CWM: Relationales Teilmodell

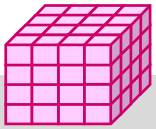




Metadaten: Architekturalternativen

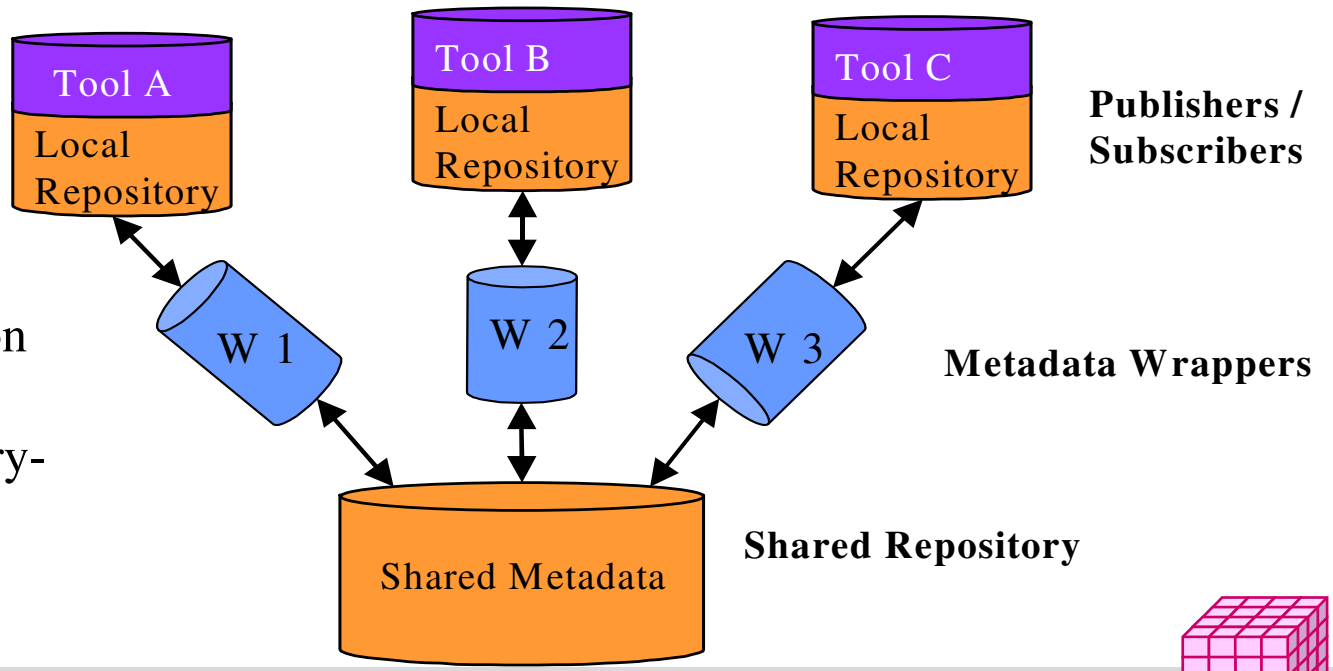


- zentrales Repository
  - keine Metadaten-Replikation
  - Abhängigkeit zu zentralem Repository auch für lokale Metadaten
  - unzureichende Autonomie
- verteilte Repositories
  - maximale Unabhängigkeit
  - schneller Zugriff auf lokale Metadaten
  - zahlreiche Verbindungen zum Metadatenaustausch
  - großer Grad an Metadaten-Replikation
  - schwierige Synchronisation
- föderiert (shared repository)
  - einheitliche Repräsentation gemeinsamer Metadaten
  - lokale Autonomie
  - begrenzter Umfang an Metadaten-Austausch
  - kontrollierte Redundanz



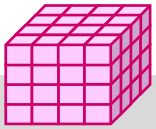
Interoperabilitätsmechanismen

- Dateiaustausch
  - kein Repository-Zugriff
  - plattform-unabhängig, einfach realisierbar , asynchron
  - Standardformate: MDIS, CDIF, XML
- Application Programming Interface (API)
  - direkter Repository-Zugriff, synchron
  - derzeit proprietär und aufwendig zu nutzen
  - Standards für Daten- und Metadatenzugriff: ODBC, OLEDB for OLAP
- Metadaten-Wrapper
  - Abbildung zwischen unterschiedlichen Metadaten-Repräsentationen
  - entweder asynchron (Dateiaustausch) oder synchron (API-basiert)
  - nur proprietäre, tool-/repository-spezifische Produkte, Anbieterabhängigkeit
  - Beispiele: Ardent MetaBroker



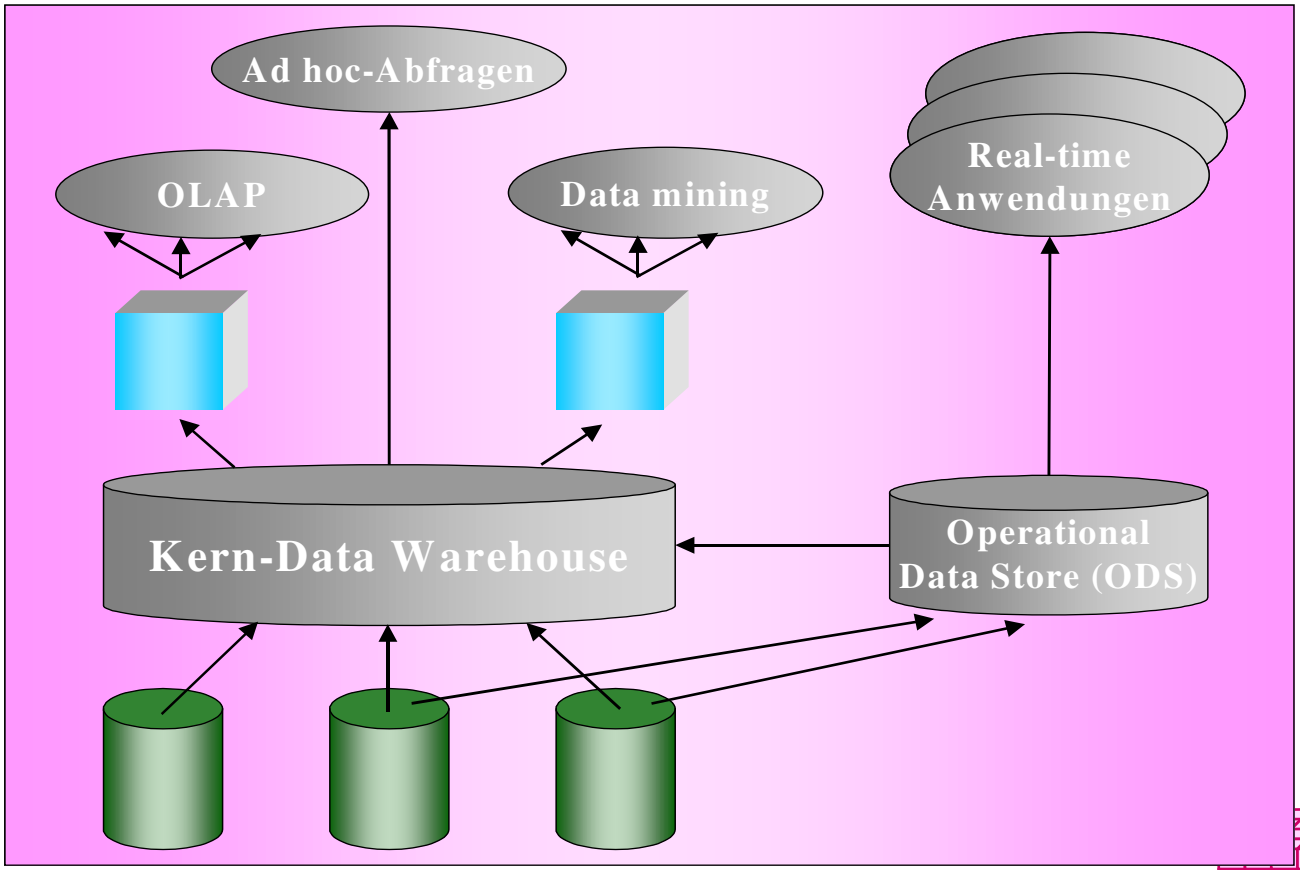
Kommerzielle Repository-Produkte

- Tool-spezifische Repositories:
  - ETL-Tools: Informatica PowerMart, PowerCenter, ...
  - Modellierungs-Tools: Sybase PowerDesigner, Oracle Designer, CA Erwin ...
- "Generische" Repositories
  - flexible und erweiterbare Metadatenmodelle, breitere Einsatzgebiete, Tool-Integration
  - Bsp.: IBM DataGuide, Microsoft Repository, Sybase, UniSys Universal Repository
- Meist proprietäre Metadaten-Modelle (realisiert über interne Datenbank) und eingeschränkte Interoperabilität
  - Import: Schema-Metadaten aus CASE-Tools / DBMS; Transformationsmetadaten aus ETL-Tools
  - Export: Querying-, Reporting- und OLAP-Tools
- v.a. passive Nutzung von Metadaten (Systemdokumentation, Nutzerinformation)
  - keine Query-Übersetzung zwischen Business-Terms und Datenbankschemata
  - kaum Unterstützung für Metadaten-/ Schemaintegration und automatische Metadaten-Synchronisation



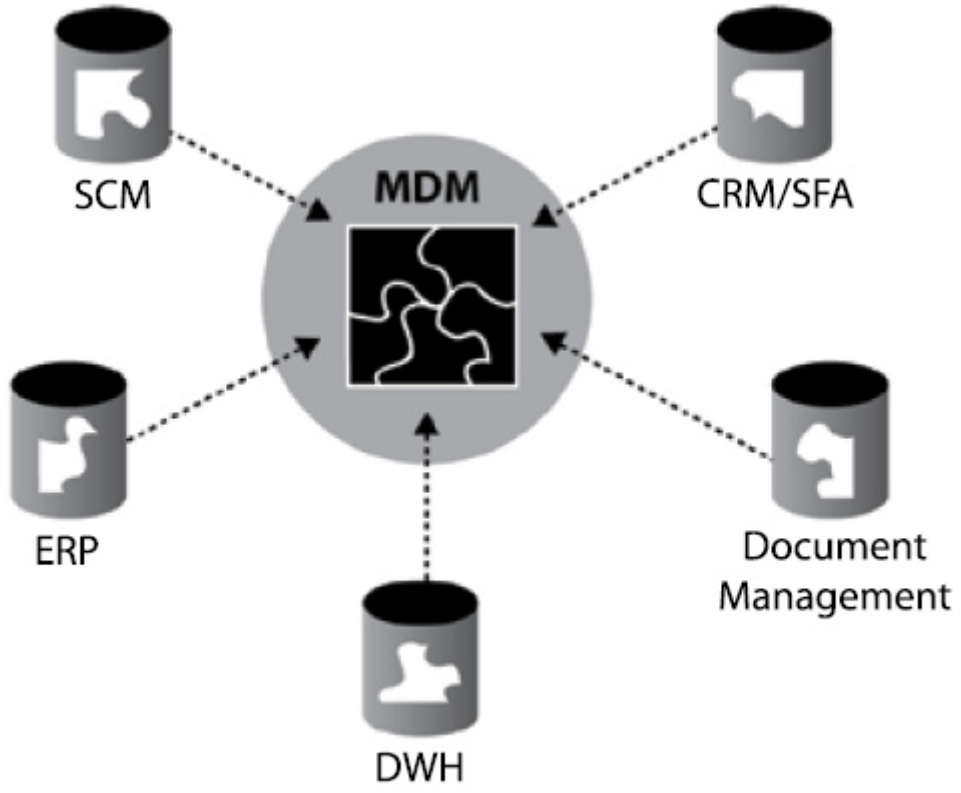
Operational Data Store (ODS)

- optionale Komponente einer DW-Architektur zur Unterstützung operativer (Realzeit-) Anwendungen auf integrierten Daten
  - grössere Datenaktualität als Warehouse
  - direkte Änderbarkeit der Daten
  - geringere Verdichtung/Aggregation, da keine primäre Ausrichtung auf Analysezwecke
- Probleme
  - weitere Erhöhung der Redundanz
  - geänderte Daten im ODS

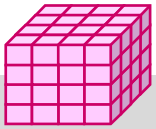


Master Data Management (MDM)

- Nutzung integrierter Masterdaten (Referenzdaten, Stammdaten) nicht nur für Analysezwecke, sondern auch für operative Anwendungen und Geschäftsprozesse
  - CDI: Customer Data Integration
  - Produktdaten, Konten, Mitarbeiterdaten, …
- MDM-Erstellung ähnelt DWH-Erstellung, jedoch unterschiedliche Nutzungsrollen
  - Replikation (Caching) von Masterdaten in Anwendungen mit Änderungsmöglichkeit
- MDM-Unterstützung im Rahmen von Anwendungsarchitekturen (SOA), z.B.
  - SAP NetWeaver , IBM , Oracle , Microsoft ….
- MDM muß skalierbar und erweiterbar sein

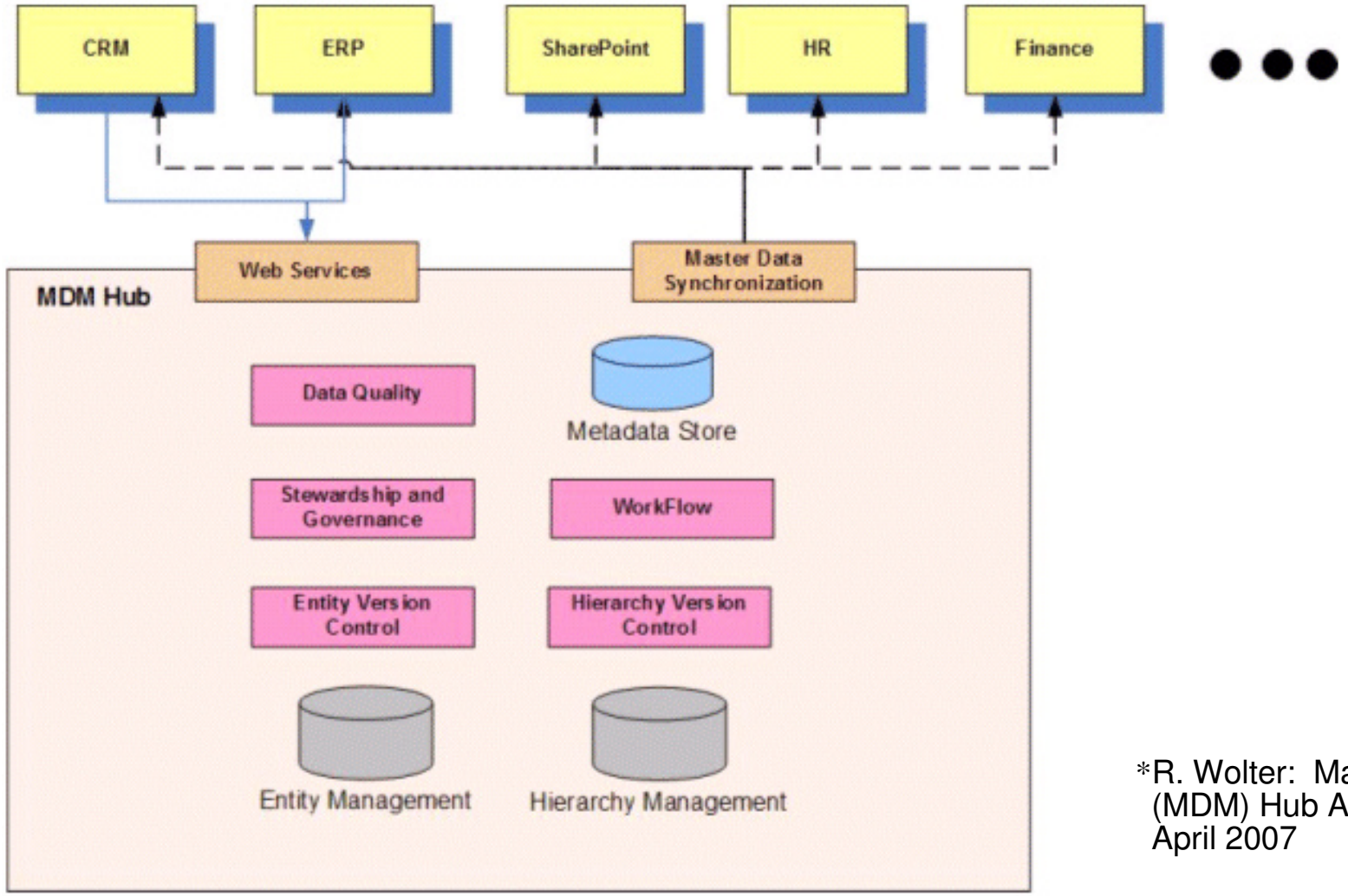


Quelle: IBM

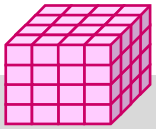


MDM-Architektur

- Materialisierte oder virtuelle Realsierung eines MDM-Hubs
- Beispiel einer Hub-Architektur (Quelle: Microsoft*)



*R. Wolter: Master Data Management (MDM) Hub Architecture. MSDN Article, April 2007



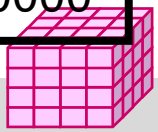
Column Stores

- DB(DW)-Inhalte werden spaltenweise statt zeilenweise abgespeichert
- Vorteile
  - Effizientere Aggregatberechnung über viele Tupel bei wenig Spalten
  - Effizientere Änderung aller Werte einer Spalte (keine Berücksichtigung anderer Spalten)
- Nachteile
  - Ineffizient falls viele Attribute eines Tupels benötigt/abgefragt werden
  - Ineffizient beim Einfügen neuer Tupel (da alle Spalten betroffen)
- Beispiel

PNr	Vorname	Nachname	Gehalt
1	Frank	Müller	30.000
2	Helga	Meier	20.000
3	Peter	Schmidt	10.000

Zeilenorientiert: 1,Frank,Müller,30000;2,Helga,Meier,20000;3,Peter,Schmidt,10000

Spaltenorientiert: 1,2,3;Frank,Helga,Peter;Müller,Meier,Schmidt;30000,20000,10000

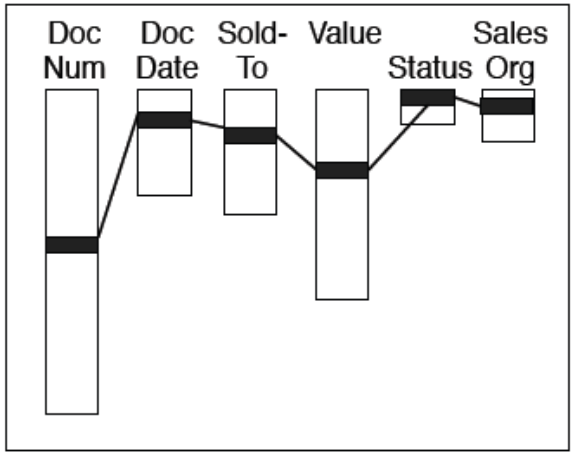


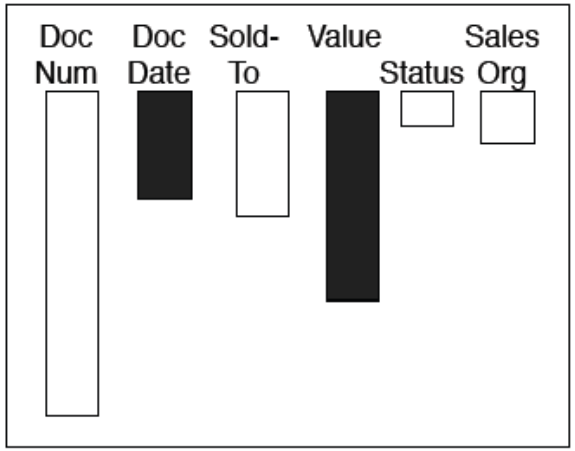
Anfragen – Column vs. Row Store *

SELECT *
FROM Sales Orders
WHERE Document Number = '95779216'

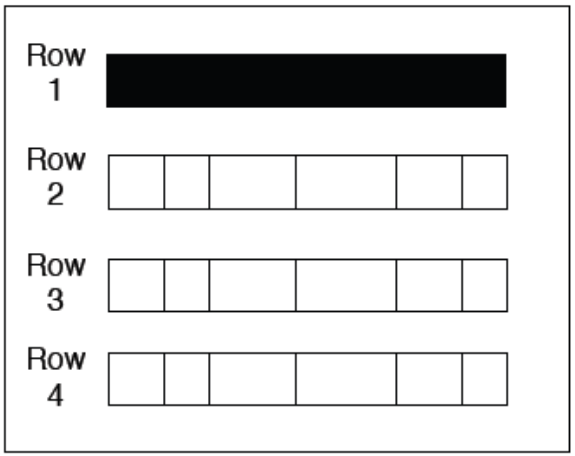
SELECT SUM(Order Value)
FROM Sales Orders
WHERE Document Date > 2009-01-20

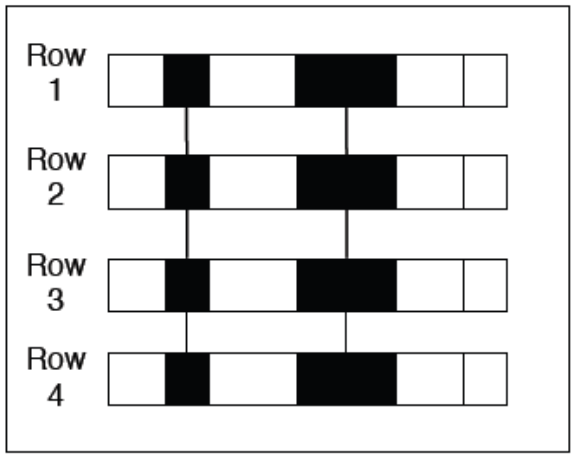
Column Store



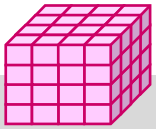


Row Store

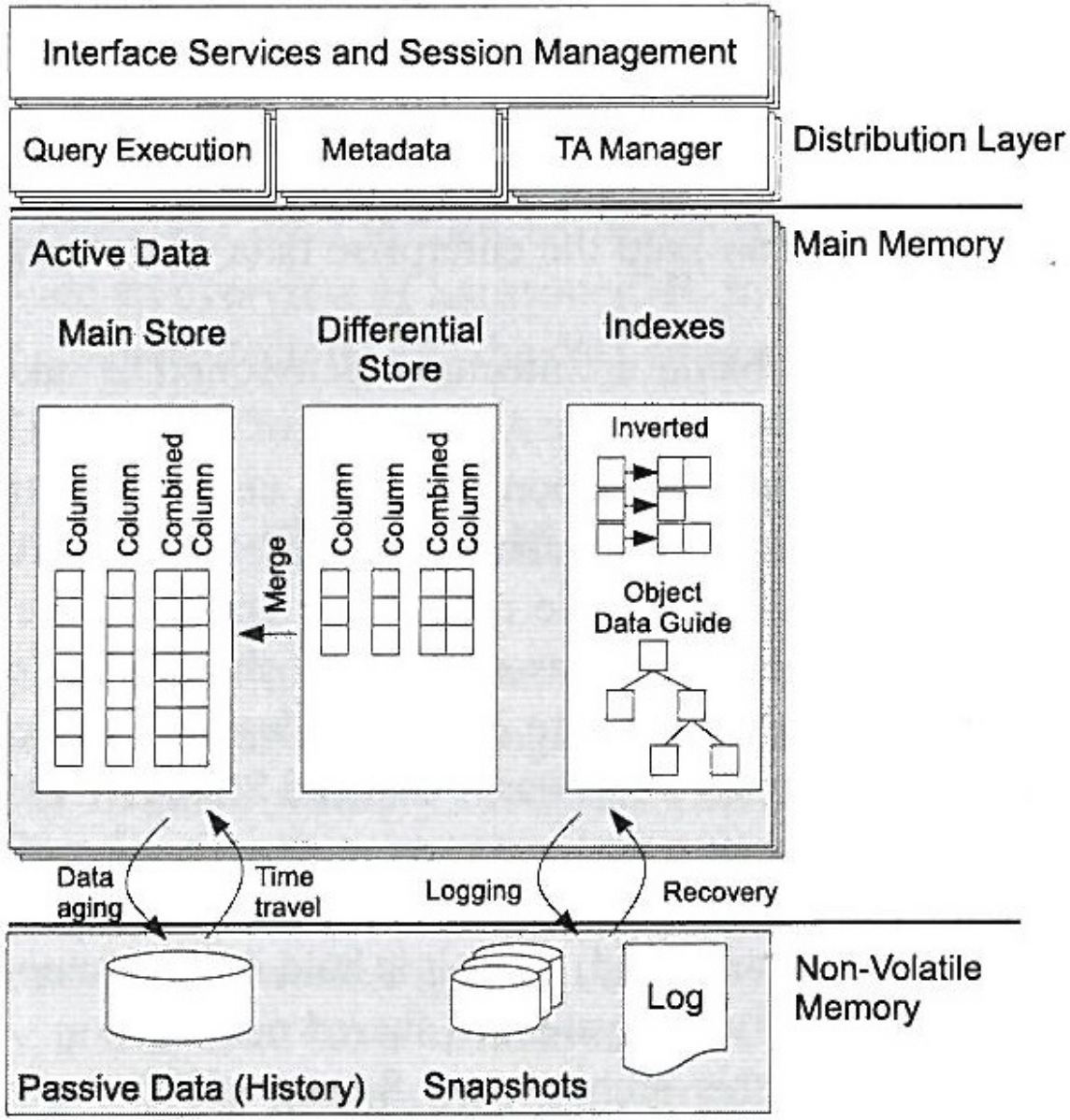




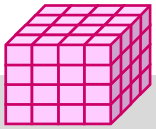
* Quelle: Hasso Plattner: Enterprise Applications – OLTP and OLAP – Share One Database Architecture



SanssouciDB *



* Quelle: Hasso Plattner, Alexander Zeier: In-Memory Data Management. An Inflection Point for Enterprise Applications



Zusammenfassung

- Komponenten der Referenzarchitektur
  - Datenquellen
  - ETL-Komponenten (Extraktion, Transformation, Laden) inklusive Monitoring und Scheduling
  - Arbeitsbereich (staging area)
  - Data Warehouse und Data Marts
  - Analyse-Tools
  - Metadaten-Verwaltung
- Extraktionsansätze: Snapshot, Trigger, Log-Transfer, DBMS-Replikationsverfahren
- Abhängige vs. unabhängige Data Marts
- Systematische Verwaltung von DW-Metadaten notwendig
  - Technische Metadaten vs. Business Metadaten, ...
  - derzeit: Ko-Existenz lokaler Repositorien mit proprietären Metadaten-Modellen
  - CWM-Standard: UML-basiert, umfassend, unzureichende Produktunterstützung
  - Metadaten-Interoperabilität v.a. über Dateiaustausch und Low-Level Repository APIs
- Unterstützung operativer Anwendungen auf integrierten Daten
  - ODS: Online Data Store
  - MDM: Master Data Management
- Column Stores