

Wir bauen uns ein Data Warehouse mit MySQL

**FrOSCon 2012, 26. August
Hochschule Bonn-Rhein-Sieg**

Oli Sennhauser

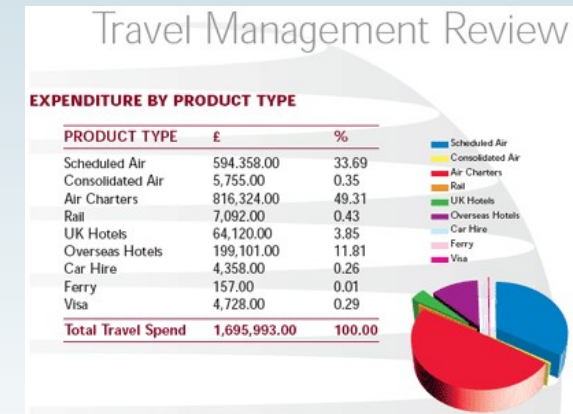
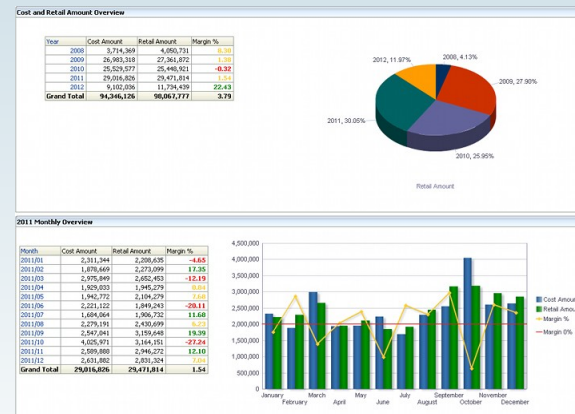
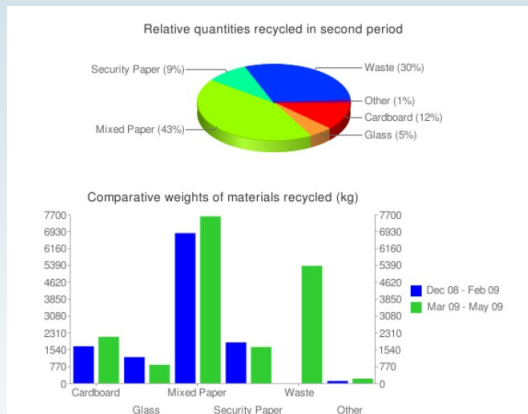
Senior MySQL Consultant, FromDual GmbH

oli.sennhauser@fromdual.com

Was ist ein Data Warehouse?

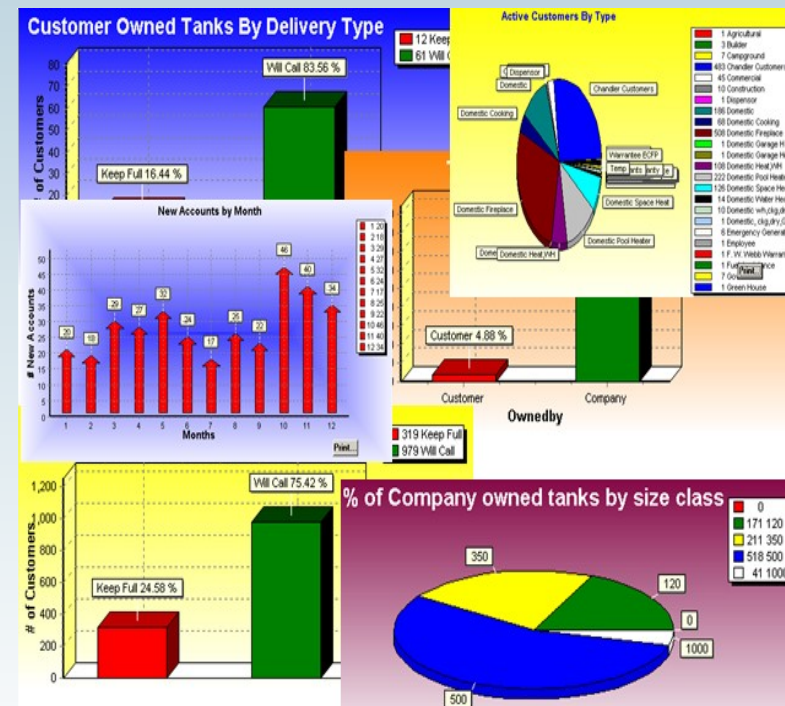
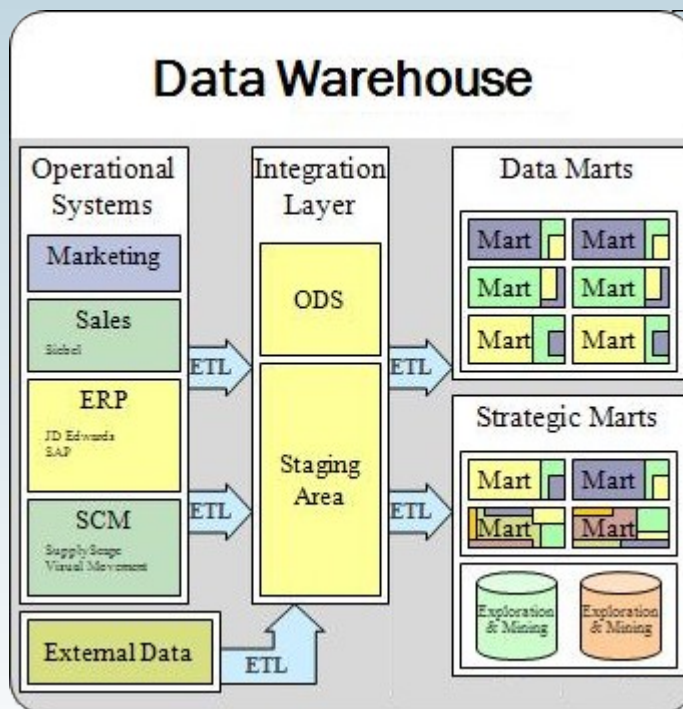
www.fromdual.com

- Data Warehouse (Abk. DWH)
 - Datenbank :-)
 - Für Reporting und Datenanalyse
 - Meist fürs Management (→ politisch, vereinfacht!)
 - Über: Finanzkennzahlen, Produktionskennzahlen, etc.
- Resultat:



Technische Sicht

- Datenquellen (OLTP System, Dateien, Fremdsysteme)
- Datenaufbereitung (ETL)
- Datenhaltung (DWH: Staging Area)
- Datenauslieferung (DWH: Data Mart)



Grober Ablauf

- **Datenquellen**
 - Operative Datenbanken: OLTP (ODBC, JDBC)
 - MySQL, Oracle, PostgreSQL, etc.
 - Dateien (CSV, XML, ...)
 - Apache Log, Host-Systeme
- **Mit ETL-Jobs (Extract, Transform, Load)**
 - Skripte (PHP, Perl, ...)
 - ETL Software (Pentaho, ...)
- **Ins DWH**
 - Zuerst Staging Area
 - Dann Data Mart (Dimensional Model, Star Schema)
- **Report (SQL, Spreadsheet, Reporting-Software)**

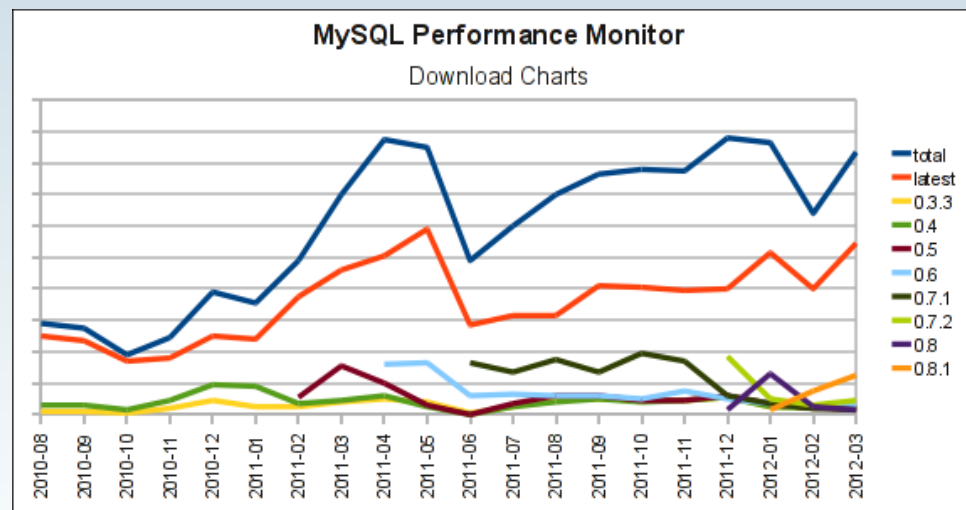
Grobes Vorgehen

Ich persönlich würde:

- **Den Gaul von hinten her aufzäumen:**
 - Welchen Report brauche ich (= Z)?
 - Woher kommen die Daten dazu (= A)?
 - Wie komme ich von A nach Z?
- **Datenfriedhöfe: Wir sammeln mal, man kann ja nie wissen...**

Beispiel

- Chef will:
 - Report über Download-Statistik unserer Produkte
- Also folgenden Report (Mock-up):
 - „Welchen Report brauche ich?“ (= Z)
 - Download Charts **nach** Produkt, Version und Zeit!



Woher die Daten?

- **Download Statistiken (= A):**
 - **Webserver Accesslog (access.log) → ~ CSV Datei**
 - **Verteilt auf mehreren Servern?**
 - **Was ist mit der Historie?**
 - **Produkt, Version und Zeit?**
- **Beteiligte Parteien/Rollen?**
 - **Kunde (= Management), SysAdmin (Server, Backup), Software Entwicklung (Produkt, Version), DWH Entwickler, ERP, Marketing, CRM, DBA, ...**

ETL Jobs

- **ETL: Extract – Transform – Load**
 - Extrahieren der Daten von externen Quellen
 - Transformieren der Daten entsprechend des Bedarfs
 - Laden der Daten ins Ziel (Staging Area oder DM)
- **Wie?**
 - Skripte (PHP, Perl, Shell, ...)
 - Software (Pentaho, ...)
- **Warum überhaupt ETL?**
 - Ich habe doch schon alle Daten in meiner DB!
 - Stimmt! Aber...

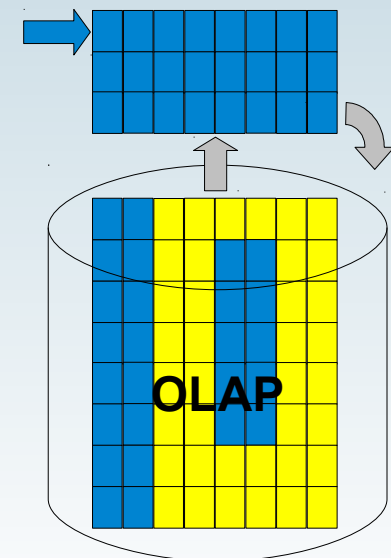
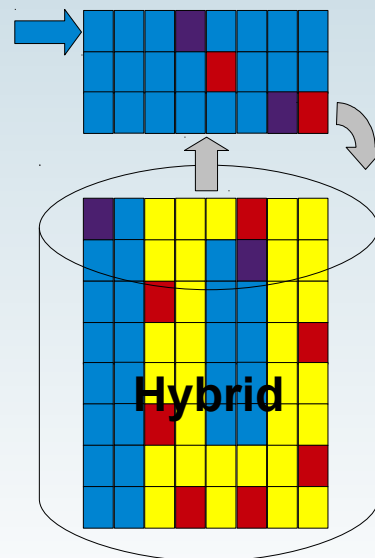
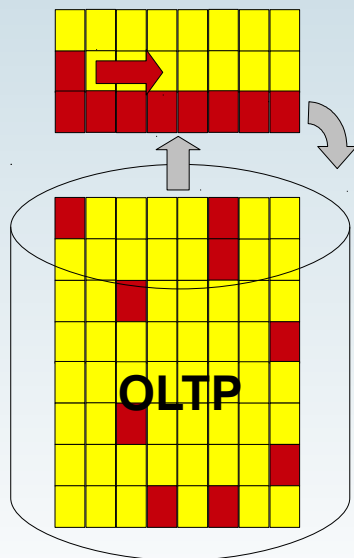
OLTP vs. OLAP

Unterschiedliche Anwendungsmuster:

- **Operative DB**
 - Online Transaction Processing (OLTP)
 - Beispiel: Webshop, ERP, VoIP Calls, etc.
 - Kurze, schnelle Zugriffe auf kleine Datenmengen
- **Reporting DB**
 - Online Analytical Processing (OLAP)
 - Beispiel: Monatsrechnungen, Umsatzentwicklung, etc.
 - Langsamere Zugriffe auf grosse Datenmengen
- **Hybride (Web-Anwendungen) :-)**
 - „Zeig mir die häufigst verkauften Produkte!“
 - „Was könnte sonst noch zu mir passen?“

Zugriffsmuster

- DB's sind schnell, wenn Daten im RAM liegen:
 - OLTP kleine Datenmengen → RAM
 - OLAP grosse Datenmengen → Platte
- Was passiert bei Kombination?
 - Sieht man sehr oft!
 - Daher wenn möglich/nötig trennen!



ETL Technisch

- **Extract**

- ODBC/JDBC + SQL
- **SELECT INTO OUTFILE → CSV**
- **mysql --xml -e 'SELECT * FROM test' > test.xml**

- **Transform**

- Sortieren der Daten nach Primary Key (InnoDB)
- DB's sind grundsätzlich langsam → wenn möglich ausserhalb der DB erledigen, schauen, welche Variante schneller ist, parallelisieren!

- **Load**

- Laden in Primary Key Sortierung (InnoDB)
- Parallel laden
- **Single Row INSERT vs multi Row INSERT vs LOAD DATA INFILE**

Laden MyISAM vs. InnoDB

- MyISAM, (PK Sortierung), keine Trx
197 s (105 %)
- InnoDB, PK Sortierung, **autocommit = 1**,
innodb_flush_log_at_trx_commit = 1
> 900 s (10 % der Daten) (**4300 %!**)
- **InnoDB**, PK Sortierung, mit Trx,
innodb_flush_log_at_trx_commit = 1
188 s (= 100 %)

Laden nach PK sortiert

- InnoDB, **PK Sortierung**, mit Trx,
`innodb_flush_log_at_trx_commit = 1`
188 s (= 100 %)
- InnoDB, KEINE PK Sortierung, mit Trx,
`innodb_flush_log_at_trx_commit = 1`
249 s (132 %)
- Sortierung: ca. **4 s**
 - („Datenbanken sind grundsätzlich langsam!“)

```
shell> sort -t, -k9 -n test_rnd.sql > test_sorted.sql
```

innodb_flush_log_at_trx_commit

www.fromdual.com

- InnoDB, PK Sortierung,
autocommit = 1, innodb_flush_log_at_trx_commit = 0
210 s (112 %)
- InnoDB, PK Sortierung,
autocommit = 1, innodb_flush_log_at_trx_commit = 1
> 900 s (10 % der Daten) (4300 %!)
- InnoDB, PK Sortierung,
autocommit = 1, innodb_flush_log_at_trx_commit = 2
236 s (126 %)

- InnoDB, PK Sortierung, mit Trx,
innodb_flush_log_at_trx_commit = 0
186 s (97 %)
- InnoDB, PK Sortierung, mit Trx,
innodb_flush_log_at_trx_commit = 1
188 s (= 100%)
- InnoDB, PK Sortierung, mit Trx,
innodb_flush_log_at_trx_commit = 2
186 s (97 %)

Parallel laden

- MyISAM, PK Sortierung, keine Trx, **parallel 4**
92 s (47 %, f = 2.1)
- MyISAM, PK Sortierung, keine Trx
197 s (= 100 %)
- InnoDB, PK Sortierung, mit Trx,
`innodb_flush_log_at_trx_commit = 1`
188 s (= 100%)
- InnoDB, PK Sortierung, mit Trx, **parallel 4**
62 s (33%, f = 3.0)

Single row vs multi row

- InnoDB, PK Sortierung, mit Trx, single row **INSERT**
188 s (= 100%)
- InnoDB, PK Sortierung, mit Trx, **multi row INSERT**
64 s (= 34%, f = 2.9)
- InnoDB, PK Sortierung, autocommit = 1, **LOAD DATA INFILE**
50 s (= 27%, f = 3.8)
- InnoDB, PK Sortierung, autocommit = 1, **INSERT INTO SELECT**
45 s (= 24%, f = 4.2)

Weitere Ladehilfen

- **INSERT ON DUPLICATE KEY UPDATE**
 - INSERT oder UPDATE
- **REPLACE INTO**
 - INSERT oder DELETE + INSERT
- **Federated Storage Engine?**
- **Transportable Tables? (MyISAM, oder InnoDB mit Percona Server)**
- **Lade-Jobs erneut startbar machen!**
 - Am 21. 8. laden wir Daten vom 20. 8.
 - Daten waren falsch
 - Am 22. 8. laden wir geflickten Daten vom 20. 8. und 21. 8.
 - Nochmal laden aber vorher löschen!?!

Welche Storage Engine?

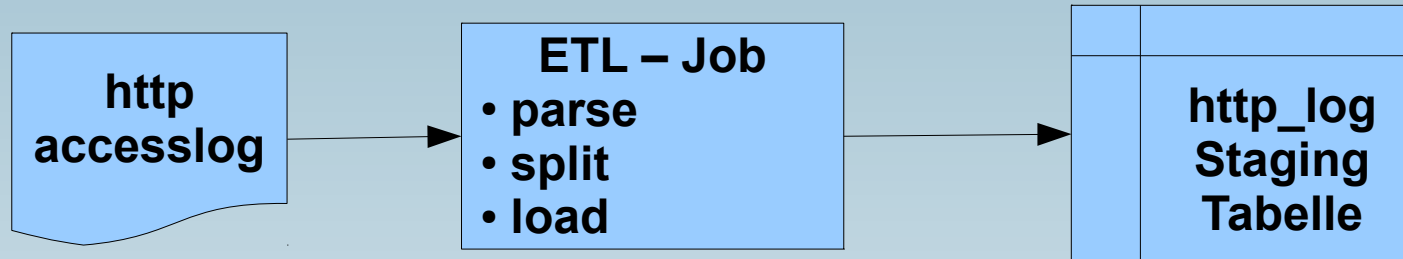
- **MyISAM (Aria)**
 - Nicht Crash-safe (Ausnahme: Aria!)
 - Table-Lock, schlecht bei Concurrency
 - Kleiner Footprint, schnell bei Full Table Scan
- **InnoDB (XtraDB)**
 - Crash-safe
 - Row-Lock, gut bei Concurrency
 - Footprint ca. 2 x MyISAM
- **Infobright, InfiniDB, Tokutek**
 - Spezialisiert (Columnar Storage Engine, Fractal Trees)
 - Proprietär, teuer
- **Spider SE?**

MySQL Konfiguration

- InnoDB:
 - `innodb_buffer_pool_size` (RAM!)
 - `innodb_log_file_size` (Data load)
 - `innodb_flush_log_at_trx_commit` (Data load)
- MyISAM:
 - `key_buffer_size` (RAM!)
 - `bulk_insert_buffer_size` (Bulk data load)
- Query Cache?
- MyISAM/InnoDB Tabellenkompression?

Staging Area

- Zurück zum Beispiel Download-Statistik:



- Data Staging Area / Tabelle

- Noch kein Report möglich...

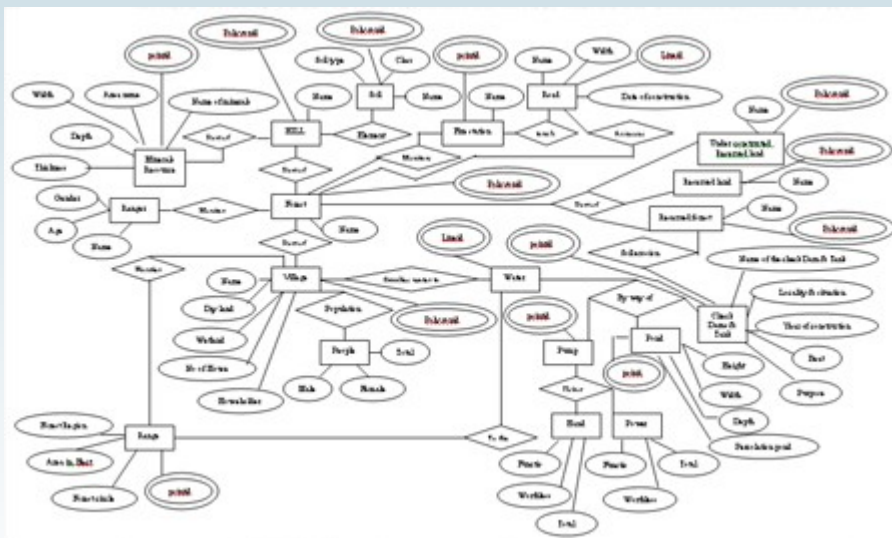
```

CREATE TABLE `page_hits_raw` (
  `host` varchar(32) DEFAULT NULL,
  `remote_host` char(15) DEFAULT NULL,
  `logname` varchar(64) DEFAULT NULL,
  `user` varchar(64) DEFAULT NULL,
  `ts` int(10) unsigned DEFAULT NULL,
  `methode` varchar(12) DEFAULT NULL,
  `request` varchar(1024) DEFAULT NULL,
  `protocol` varchar(16) DEFAULT NULL,
  `status` smallint(5) unsigned DEFAULT NULL,
  `bytes` mediumint(8) unsigned DEFAULT NULL,
  `referer` varchar(1024) DEFAULT NULL,
  `user_agent` varchar(255) DEFAULT NULL,
  `ua_engine_compatibility` varchar(255) DEFAULT NULL,
  `ua_operating_system` varchar(255) DEFAULT NULL,
  `ua_platform` varchar(255) DEFAULT NULL,
  `ua_browser_platform_details` varchar(255) DEFAULT NULL,
  `ua_browser_enhancements` varchar(255) DEFAULT NULL
);
  
```

Relational vs. Dimensional

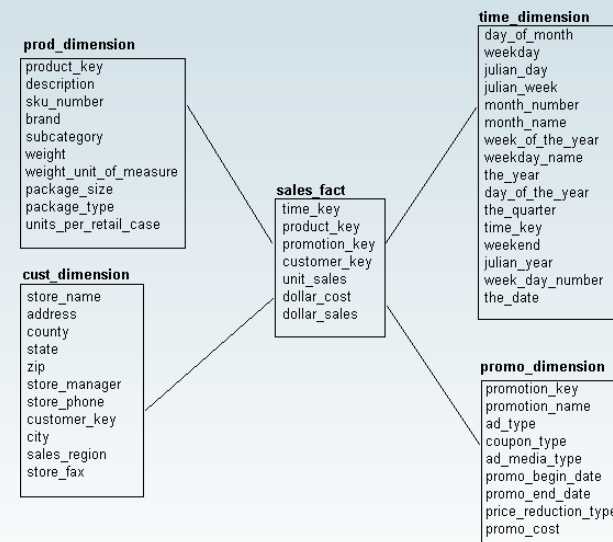
• OLTP

- Entity Relationship Model (ER Schema)
- Normalisiert (3rd NF)
- Keine Redundanz
- Komplexes Schema
- Vielen Tabellen und Joins
- Hohe Schreib-Performance



• OLAP

- Dimensional Model (Star Schema)
- Stark denormalisiert
- Hohe Redundanz
- Einfaches Schema
- Wenige Tabellen und Joins (Joins sind teuer!)
- Hohe Lese-Performance

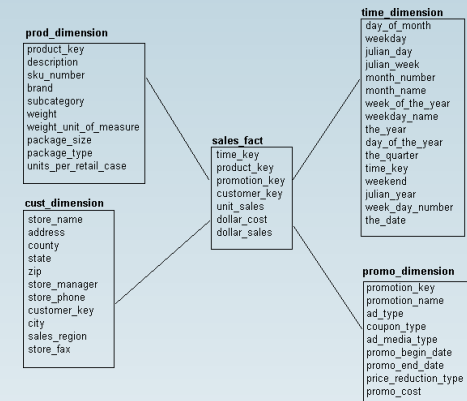
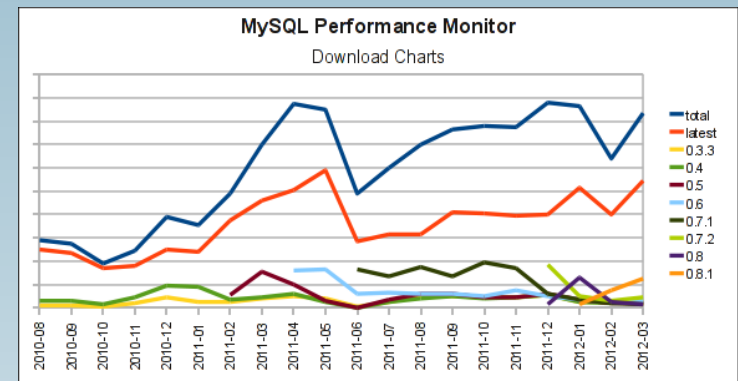


Dimensional Schema Design

www.fromdual.com

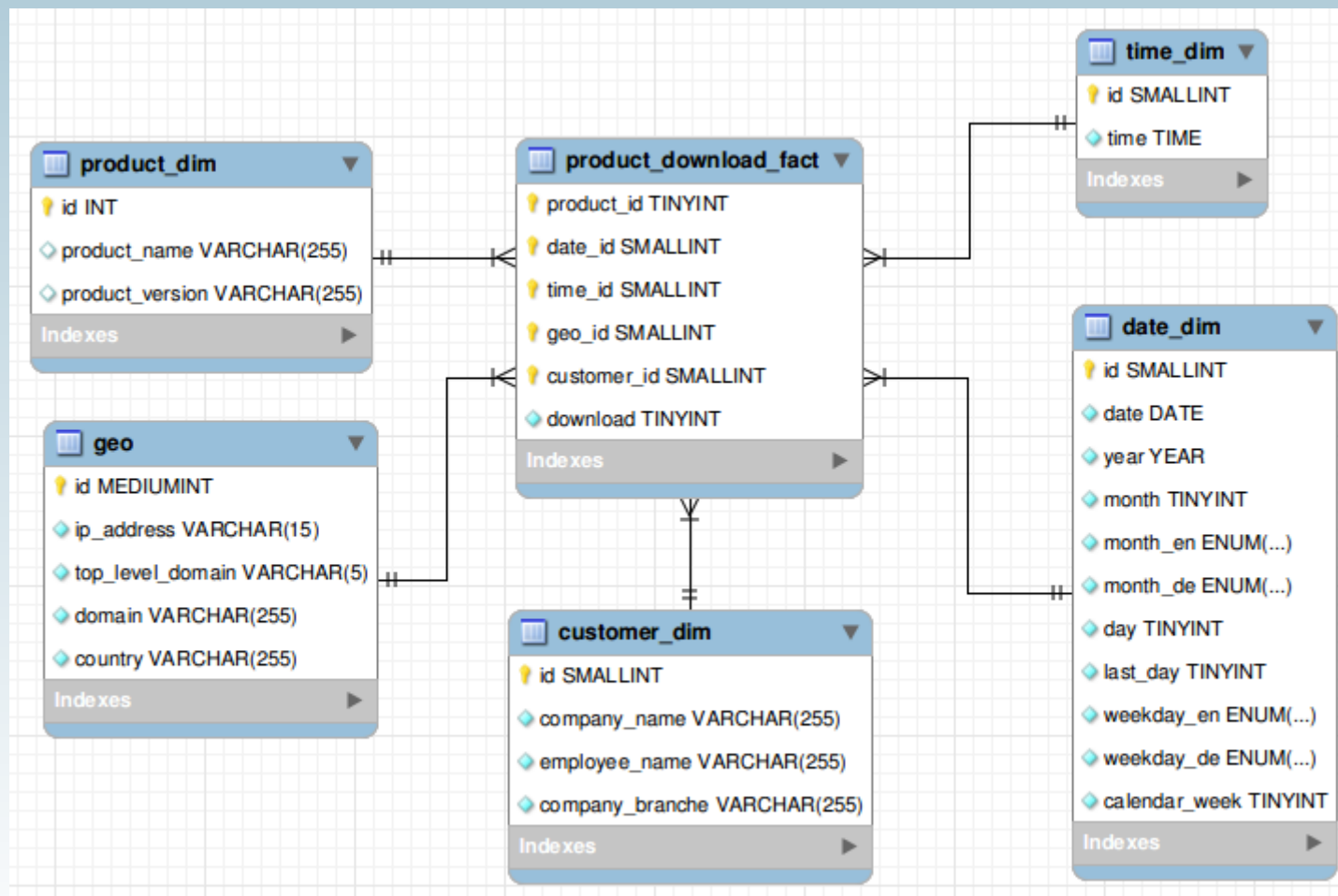
Star Schema:

- Fact Tabellen
 - Beinhaltet die Messwerte (Downloads der Produkte)
- Dimension Tabellen
 - Einstiegspunkt für Fact Tabellen
 - Beschreibung des Geschäftsprozesses
 - Oft „sortieren/gruppieren **nach** Produkt, **nach** Zeit, **nach** Version etc...”



Star Schema

- „Downloads sortieren/gruppieren **nach** Produkt, **nach** Zeit und **nach** Version!“



Fact Tabellen

- Speichert numerisch Performance-Metriken
- Werte sind additiv (1 download + 1 download)
- Möglichst atomar (NICHT 7 downloads pro Tag!)
- Selbe Granularität über ganze Tabelle (sonst Aggregate)
- Wenig Spalten (Anz. Dim + 2 – 5 Facts) aber oft sehr viele Zeilen! → Grosse Tabellen, daher auf Datentypen achten!
- Lokalität der Daten beeinflussbar:
 - InnoDB Primary Key → physische Sortierung der Zeilen
 - MyISAM: `ALTER TABLE ... ORDER BY ...` (teuer!)
 - Partitionen (RANGE, meist nach Zeit)!
- Auf Dimension-Bus-Architektur passend

Fact Beispiel

```
CREATE TABLE `product_download_fact` (
  `product_id` tinyint(4) NOT NULL,
  `date_id` smallint(6) NOT NULL,
  `time_id` smallint(6) NOT NULL,
  `geo_id` smallint(6) NOT NULL,
  `customer_id` smallint(6) NOT NULL,
  `download` tinyint(4) NOT NULL,
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`id`),
  KEY `product_id` (`product_id`),
  KEY `date_id` (`date_id`),
  KEY `time_id` (`time_id`),
  KEY `geo_id` (`geo_id`)
);
```

product_id	date_id	time_id	geo_id	customer_id	download	id
0	9483	1389	36	57	1	1
0	9487	1395	36	79	1	2
0	9489	1396	37	85	1	3
0	9492	1376	34	5	1	4
0	9492	1414	39	0	1	5
0	9493	1365	32	0	1	6
0	9494	1382	34	28	1	7
1	2	1283	20	0	1	8
1	7	1316	25	23	1	9
1	8	1329	27	71	1	10

Dimension Tabellen

- Einstieg und User Interface ins DWH (auf Fact Tabellen)
- Gehören zu Fact Tabellen
- Textbeschreiben des Geschäfts
- Hoch redundant
- Oft „sortieren/gruppieren **nach** Produkt, **nach** Zeit, **nach** Version etc...”
- Haben viele und lange Spalten (50 – 100, hoch redundant)
 - MySQL Limitationen beachten!
- Üblicherweise nur wenige Zeilen (< 1 Mio)
- Hier findet Einschränkung (**WHERE**), Gruppierung (**GROUP BY**) und Sortierung (**ORDER BY**) statt
- Verwendung in Report-Titeln
- Gute Hierarchie ermöglicht „Slice and Dice“ („Scheibchen und Würfel schneiden“)
- Abkürzungen und **NULL** Werte sind verpönt
- Sollte in Dimensions Bus-Architektur passen

Dimension Beispiel

```
CREATE TABLE product_dim (
  id                smallint(5) unsigned NOT NULL A_I
, product_name      varchar(255) NOT NULL
, product_version   varchar(255) NOT NULL
, PRIMARY KEY (id)
);
```

id	product_name	product_version
0	Unknown	Unknown
1	MySQL Performance Monitor	Unknown
2	MySQL Performance Monitor	latest
11	MySQL Performance Monitor	0.9
12	Database Control	Unknown
13	Database Control	latest
14	Database Control	0.1
15	MyEnv	Unknown
16	MyEnv	latest
20	MyEnv	0.5
21	Nagios Plugins	Unknown
25	Nagios Plugins	0.3.1

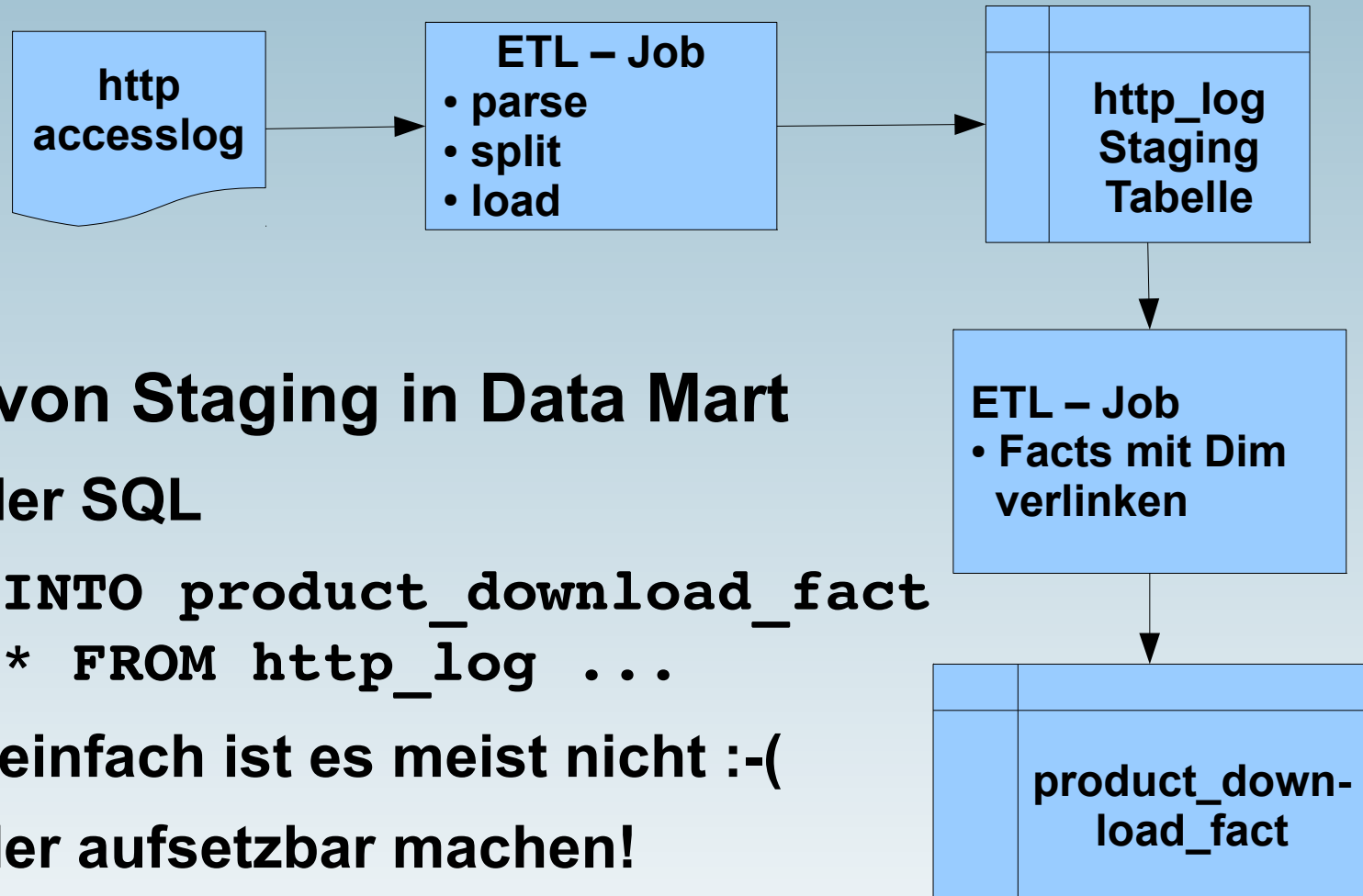
```
CREATE TABLE date_dim (
  id                smallint(5) unsigned NOT NULL A_I
, date              date NOT NULL
, year              year(4) NOT NULL
, month             tinyint(3) unsigned NOT NULL
, month_en          enum('January','February'...
, month_de          enum('Januar','Februar'...
, day               tinyint(3) unsigned NOT NULL
, last_day          tinyint(3) unsigned NOT NULL
, weekday_en        enum('Monday','Tuesday','Wednesday'...
, weekday_de        enum('Montag','Dienstag','Mittwoch'...
, calendar_week     tinyint(3) unsigned NOT NULL
, PRIMARY KEY (id)
);
```

id	date	year	month	month_en	day	last_day	weekday_en	calendar_week
4597	2012-08-01	2012	8	August	1	30	Wednesday	31
4598	2012-08-02	2012	8	August	2	29	Thursday	31
4599	2012-08-03	2012	8	August	3	28	Friday	31
4600	2012-08-04	2012	8	August	4	27	Saturday	31
4601	2012-08-05	2012	8	August	5	26	Sunday	31
4602	2012-08-06	2012	8	August	6	25	Monday	32
4603	2012-08-07	2012	8	August	7	24	Tuesday	32
4604	2012-08-08	2012	8	August	8	23	Wednesday	32
4605	2012-08-09	2012	8	August	9	22	Thursday	32
4606	2012-08-10	2012	8	August	10	21	Friday	32

DWH Bus Architektur

Business Process	Business Priority	Conformed Dimensions										
		Date (Order, Start, Ship)	Product	Promotion	Customer	Employee	Page	Internet Registered User	Part	Vendor	Shipper	Problem
Orders Forecasting	2	X	X	X	X	X						
Orders	1	X	X	X	X	X						
Purchasing		X	X		X	X			X	X	X	
Parts Inventory		X	X	X					X	X		
Manufacturing	6	X	X						X			
Finished Goods Inventory		X	X	X								
Shipping		X	X	X	X	X					X	
Returns	5	X	X		X	X					X	
Registration Cards		X	X		X							
Customer Calls	4	X	X	X	X	X			X			X
Web Support		X	X		X	X						X
Financial Forecasting		X	X	X	X	X	x	x		x		
Exchange Rate Management	3	X										

ETL von Staging in Data Mart



- Kopieren von Staging in Data Mart
 - Skript oder SQL
 - `INSERT INTO product_download_fact
SELECT * FROM http_log ...`
 - Ganz so einfach ist es meist nicht :-(
 - Job wieder aufsetzbar machen!
- Fertig zum Reporting!

Slowly Changing Dimensions

- **Relativ statisch aber nicht ganz**
 - Beispiel: Kunde zieht um von A nach B
 - Report „Umsatz nach Bundesland“?
- **Type 1: Überschreiben der alten Werte**
 - Einfach aber Historie geht verloren
- **Type 2: Neuer Eintrag in Dimensionstabelle**
 - Meistgebrauchte Lösung (Historie)
- **Type 3: Neue Spalte in Dimension**
 - Report nach alt und neu möglich (Vergleich mit letztem Jahr!)

Reporting

- **SQL → Nicht ganz Manager tauglich :-)**
- **Job: `SELECT INTO OUTFILE` → Spreadsheet (OO calc)**
- **OO calc über ODBC/JDBC auf DB lassen**
- **Reporting Tools (Pentaho, ...)**

SELECT → CSV → OO calc

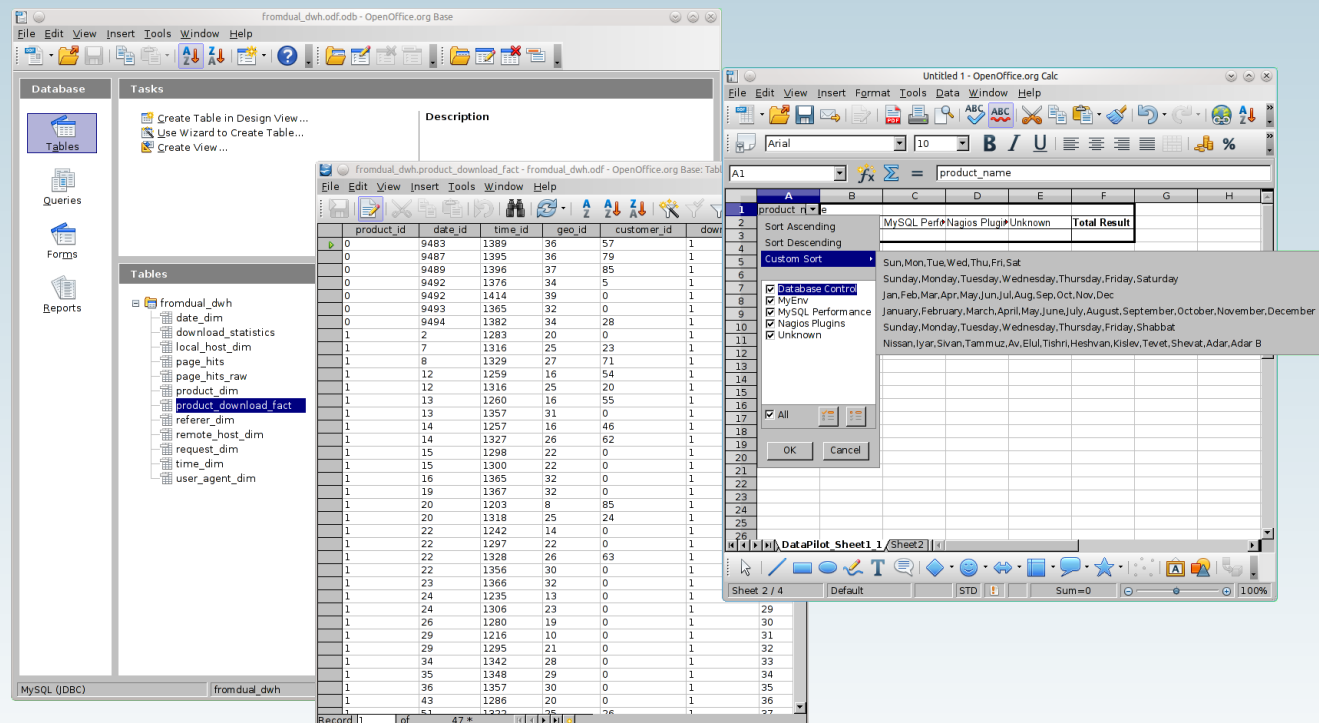
```
SELECT *
FROM date_dim AS dd
JOIN product_download_fact AS pdf ON pdf.date_id = dd.id
JOIN product_dim AS pd ON pdf.product_id = pd.id
WHERE pd.product_name = 'MySQL Performance Monitor'
AND dd.year between 2007 and 2012
INTO OUTFILE '/tmp/oli/product_download_2007-2012.csv'
;
```

1385 File Edit View Insert Format Tools Data Window Help

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	date	year	Month	Monat	Day	Last_day	Day of week	Wochentag					
10333	2768	30.07.07	2007	7 July	Juli	30	1 Monday	Montag	31	9	27		
10417	2922	31.12.07	2007	12 December	Dezember	31	1 Monday	Montag	1	9	29		
10418	2922	31.12.07	2007	12 December	Dezember	31	1 Monday	Montag	1	9	29		
10572	3195	29.09.08	2008	9 September	September	29	1 Monday	Montag	40	9	31		
10803	3622	30.11.09	2009	11 November	November	30	1 Monday	Montag	49	9	36		
11394	4749	31.12.12	2012	12 December	Dezember	31	1 Monday	Montag	1	9	47		
13855													
13856													
13857													

OO calc → JDBC auf DB

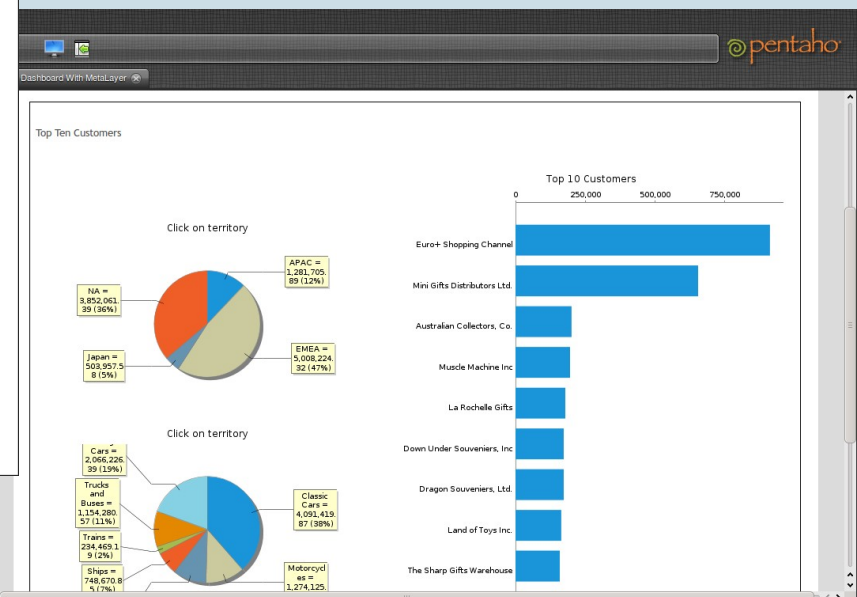
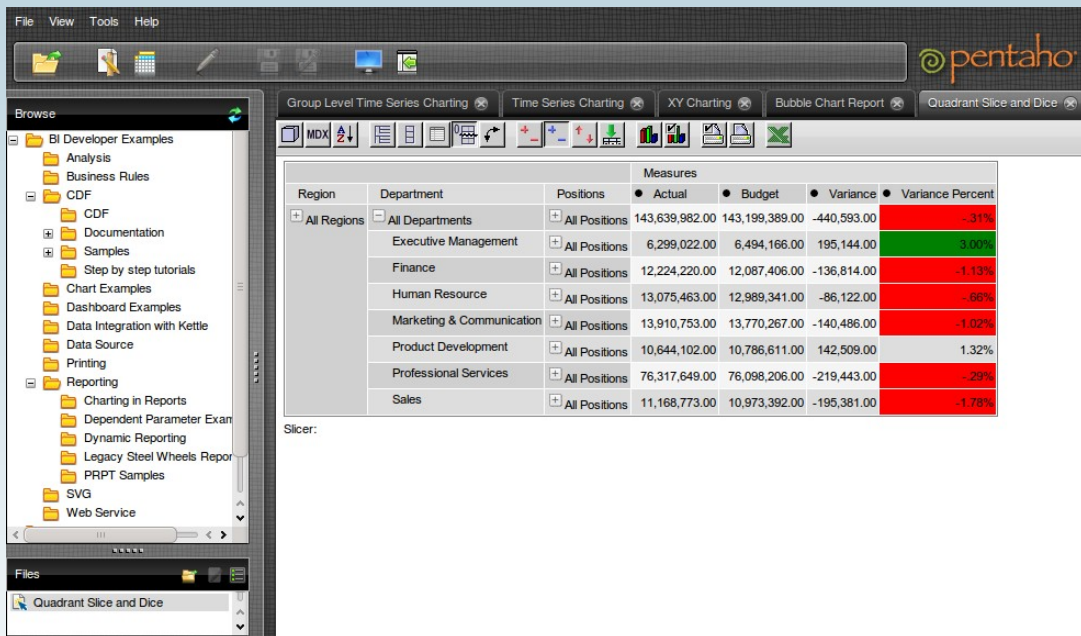
- MS Access oder MS Excel sind theoretisch auch möglich :-)
- Connector/J (JDBC)
- OO calc → gestorben :-)
- Ggf. VIEW verwenden



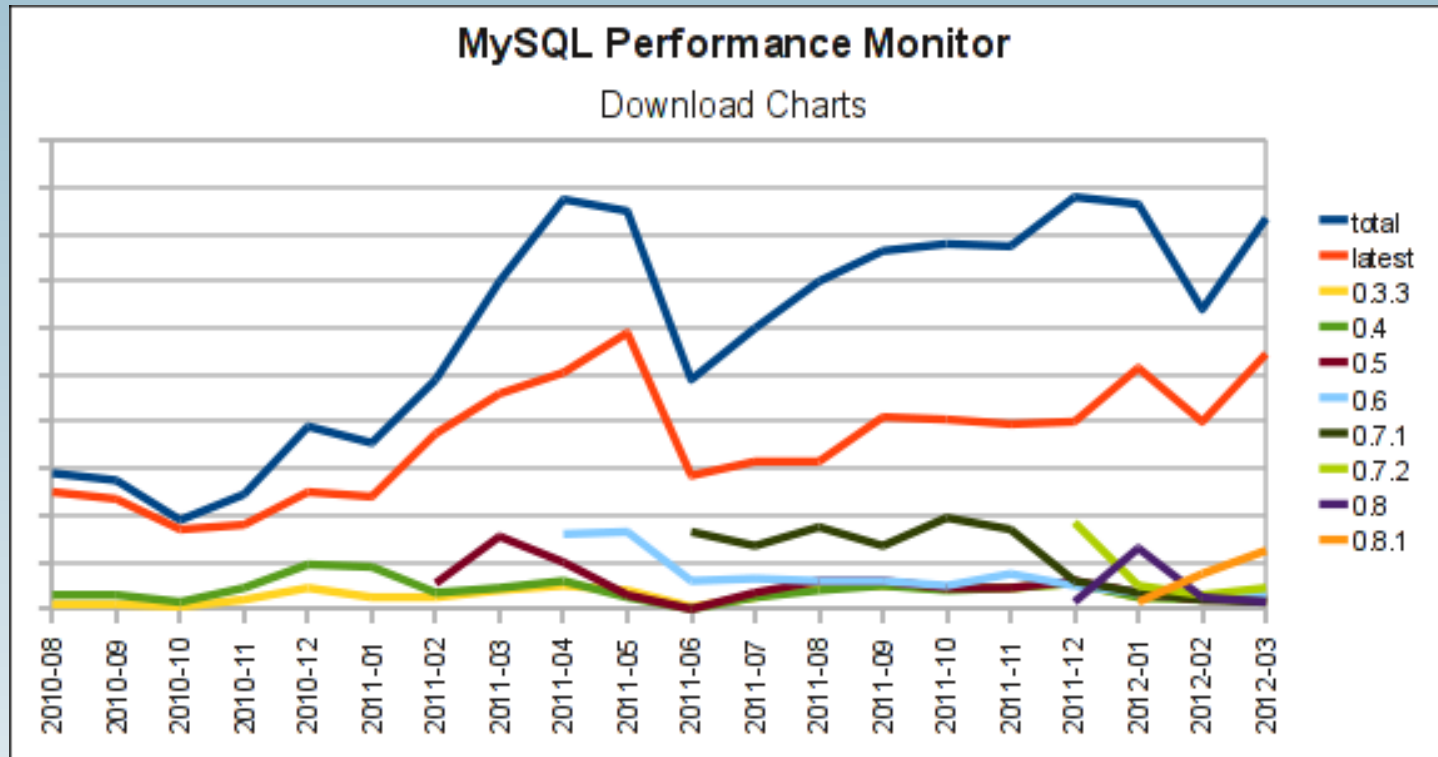
The screenshot displays two overlapping windows from the OpenOffice suite. The background window is 'fromdual_dwh.odf - OpenOffice.org Base', showing a database structure with a 'Tables' list on the left and a 'Description' pane on the right. The 'Tables' list includes 'fromdual_dwh', 'date_dim', 'download_statistics', 'local_host_dim', 'product_dim', 'product_download_fact', 'referrer_dim', 'remote_host_dim', 'request_dim', 'time_dim', and 'user_agent_dim'. The 'product_download_fact' table is selected, and its data is visible in a grid view with columns: product_id, date_id, time_id, geo_id, customer_id, and download_id. The foreground window is 'Untitled 1 - OpenOffice.org Calc', showing a spreadsheet with a data table. The table has columns A through H, with row 1 containing 'product_name' and 'Total Result'. The data in the table includes various MySQL performance metrics and Nagios plugins. A 'Sort' dialog box is open over the spreadsheet, showing options for sorting by 'product_name' in ascending order.

Reporting Tools

- Pentaho Business Analytics OSS
- JasperSoft (JasperReports) OSS
- Crystal Reports, MicroStrategy, Cognos, ...



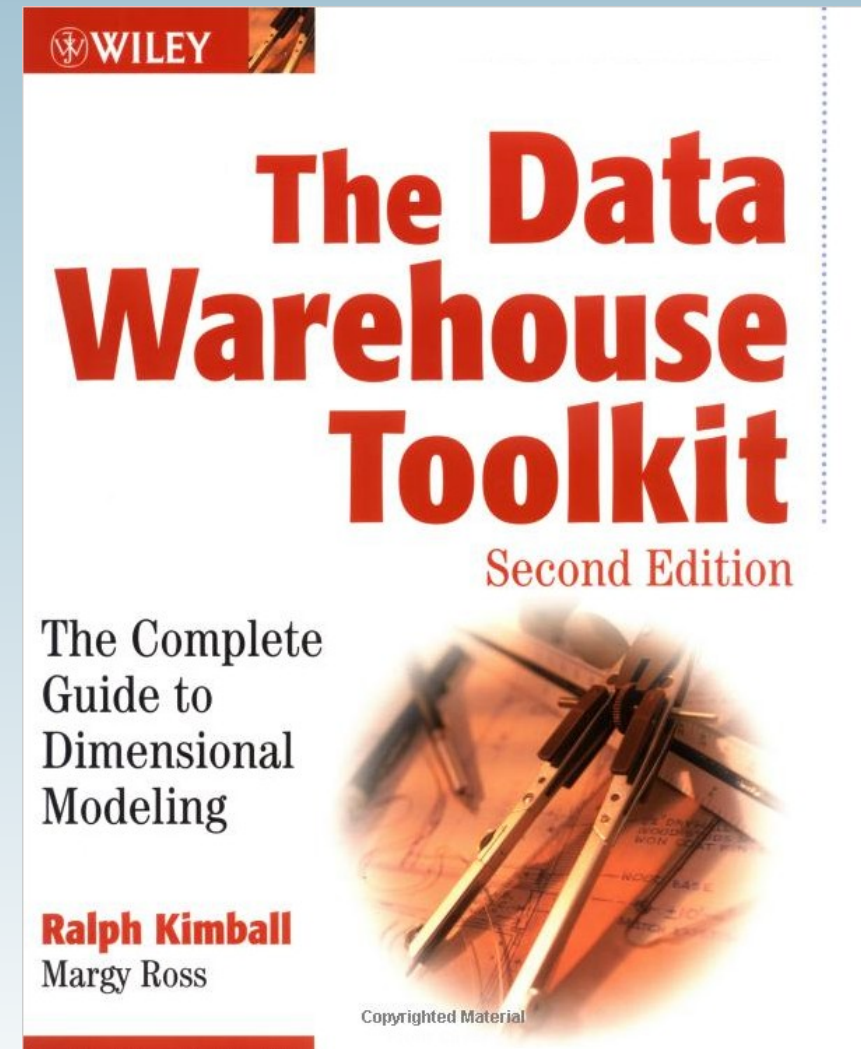
Ziel erreicht!



- Zeit: ab 6 – 8 Wochen (30 – 40 d, € 25 - 35k)

Buchtipp

- The Data Warehouse Toolkit
- Ralph Kimball & Margy Ross



Q & A



www.fromdual.com



Fragen ?

Diskussion?

Wir haben Zeit für ein persönliches Gespräch...

- **FromDual bietet neutral und unabhängig:**
 - **Beratung**
 - **Remote-DBA**
 - **Support für MySQL, Galera und Percona Server und MariaDB**
 - **Schulung**

www.fromdual.com