Wir bauen uns ein Data Warehouse mit MySQL

Oli Sennhauser
FromDual GmbH
Uster / Schweiz

Schlüsselworte
MySQL, DWH, Data Warehouse, ETL, BI, Business Intelligence, OLAP

Einleitung
MySQL Datenbanken verbreiten sich immer mehr in geschäftskritischen Bereichen. Finanzbuchhaltungen, Rechnungsstellungs-Systeme von weltweit agierenden Grosskonzernen und ERP-Systeme von mittelständischen Unternehmen basieren auf MySQL.
Bei soviel wichtigen Daten tritt früher oder später das Bedürfnis nach entsprechenden Reports auf, welche die wichtigsten Kennzahlen liefern.
Diese Reports basieren auf Zahlenmaterial, welches aus den einzelnen Teilsystemen zusammengesammelt und im Data Warehouse für die Reports bereitgestellt wird. Zum Glück müssen wir für dieses Data Warehouse nicht auf ein anderes Datenbank-System zurückgreifen sondern können diese Arbeit auch mit einer MySQL Datenbank bewältigen.
Wie das technisch umgesetzt wird beleuchten wir in diesem Vortrag.

Aufbaue eines Data Warehouses
Der Aufbau eines Data Warehouses ist ein wenig komplexer als es eine einfache Applikation mit Ihrer Datenbank ist. Der Grund hierfür ist recht einfach: Aus verschiedenen Datenquellen wie der Fi-Bu, dem CRM-System, dem ERP-System, aus dem Marketing und möglicherweise aus externen Quellen können Daten bezogen werden, welche in sogenannten ETL-Prozessen aufbereitet, bereinigt und allenfalls angereichert werden müssen. Diese Daten werden in einer sogenannten Staging Area abgelegt. Aus diesen Daten werden in einem weiteren Schritt sogenannte Data Marts und Aggregate davon gebildet.

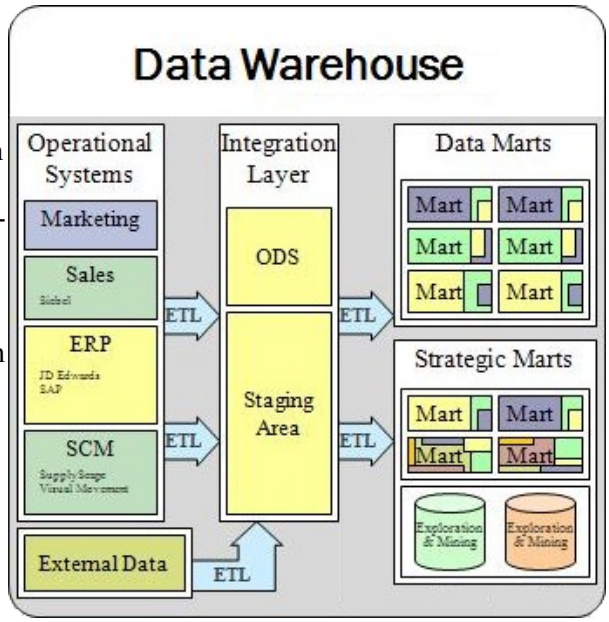

Abb. 1: Schematischer Aufbau eines Data Warehouses

Aus den Daten dieser Data Marts werden anschliessend die entsprechenden Kennzahlen zu Reports verarbeitet, welche dann für Geschäftsentscheidungen verwendet werden.

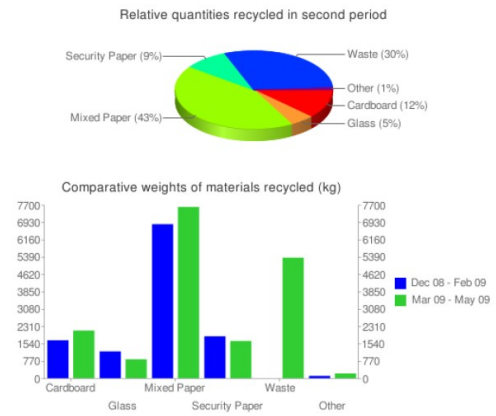

Abb. 2: Graphische Aufbereitung der Kennzahlen

Vorgehen

Damit DWH Projekte nicht im Uferlosen enden, ist es sinnvoll zuerst festzulegen, welche Reports benötigt werden und welche Kennzahlen es zu ermitteln gilt. Anschliessend kann man sich auf die Suche nach den entsprechenden Datenquellen machen und sich überlegen, wie diese angezapft und die Daten in das MySQL Data Warehouse überführt werden.
Oft findet man Systeme in welchen Daten auf Vorrat gesammelt werden ohne konkreten Nutzen, nach dem Motto: „man kann ja nie wissen“. Solche Systeme werden dann unter Umständen sehr gross und sind nicht mehr einfach zu betreiben.
Um an die Daten zu gelangen (Extrahieren) stehen grundsätzlich die folgenden Methoden zur Verfügung: Direktes ansprechen der Datenquellen mittels ODBC, JDBC oder dem MySQL Protokoll, wenn es sich bei dem Quellsystem um eine MySQL Datenbank handeln sollte oder aber das Laden der angelieferten Daten-Dateien des Quellsystems.

Transformieren und Laden der Daten nach MySQL

Um die Daten zu transformieren (filtern, bereinigen, anreichern) stehen wiederum 2 Möglichkeiten zur Verfügung. Entweder wir verlassen uns auf die Funktionalität von ETL-Tools (wie es zum Beispiel mit Pentaho mitgeliefert wird) oder wird schreiben unsere Jobs von Hand in einer beliebigen Sprache. Als grobe Regel gilt: Daten ausserhalb der Datenbank zu verarbeiten ist üblicherweise schneller als in der Datenbank. Um die volle Kapazität moderner Rechner zu nutzen und die entsprechenden Laufzeiten zu verkürzen ist es zudem Sinnvoll diese Transformierungsschritte zu parallelisieren.
Bevor die Daten geladen werden, muss man sich noch im Klaren sein, welche MySQL Storage Engine verwendet werden soll: MyISAM oder InnoDB. Beide Storage Engines haben ihre Vor- und Nachteile, wobei der Trend zunehmend in Richtung InnoDB geht.
Je nachdem wie die Daten geladen werden, können signifikante Unterschiede in er Laufzeit erreicht werden. Es können durchaus Unterschiede von Faktor 100 auftreten, je nachdem welche Methode man wählt. Besondere Vorsicht ist geboten bei einer nicht weiter konfigurierten MySQL Datenbank, InnoDB Tabellen, single Row INSERTs und Auto-commit.

Folgende Einstellungen sind förderlich für eine kürzere Ladezeit:

- Sortieren der Daten nach dem Primary Key vor dem Laden.
- Verwenden von expliziten Transaktionen um grössere Blöcke von Daten.
- `innodb_flush_log_at_trx_commit = { 0 | 2 }`
- Paralleles Laden der Daten.
- Bulk Loader Methoden verwenden (multi Row `INSERT`, `LOAD DATA INFILE`, `INSERT INTO SELECT * FROM`)

Das Dimensional Schema
Sind die Daten in eine Tabelle der Staging Area geladen kann mit diesen üblicherweise noch nicht allzu viel angefangen werden, da sie noch nicht in der richtigen Form vorliegen. Für das Design der Data Marts wird das sogenannte Dimensionale Modell im Gegensatz zum Relationalen Modell verwendet um ein sogenanntes Star-Schema zu erstellen. Der Grundgedanke hierbei liegt darin begründet, dass komplexe Join-Abfragen sehr teuer sind. Es wird durch eine sehr hohe Denormalisierung ein relativ simples Design, das sogenannte Star-Schema erreicht. Hier haben üblicherweise nur sehr simple Joins welche sich meist über 2 – 4 Tabellen erstrecken.

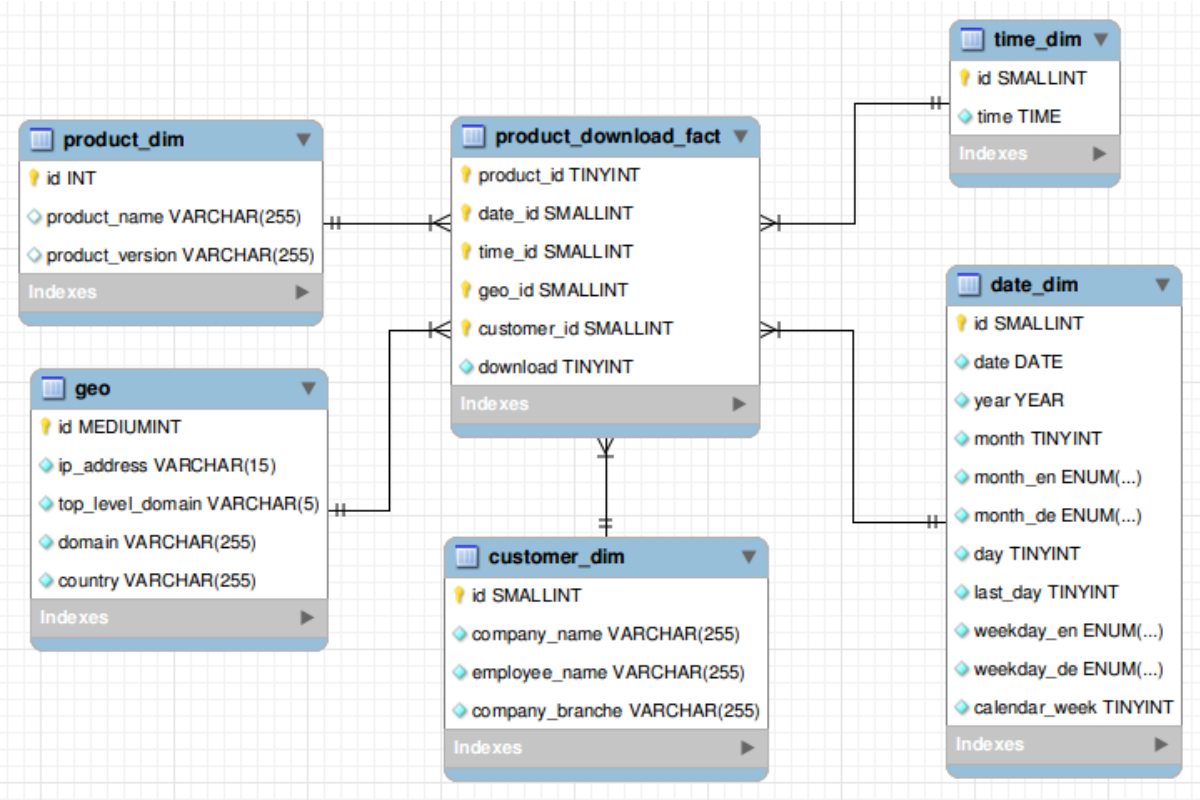

Abb. 3: Einfaches Star-Schema

In einem solchen Star-Schema sprechen wir von Fact-Tabellen, welche die eigentlichen Messwerte enthalten und den Dimensions-Tabellen welche Filterkriterien enthalten.
Diese Dimensions-Tabellen sind die typischen Einstiegspunkte für eine Abfrage. Sie dienen als Label für Reports, nach ihnen wird sortiert und gruppiert. Dimensions-Tabellen lassen sich einfach ermitteln indem man die Fragestellung anschaut: „gruppieren nach Produkt, nach Datum und nach Kunde“. Diese Fragestellung gibt uns bereits einen Hinweise welche Dimensions-Tabelle wir für das Schema verwenden werden, nämlich: Produkt, Datum und Kunde.
Bei einem guten Design passen diese Dimensions-Tabellen auch in die sogenannte DWH Bus Architektur. Die Idee dahinter ist recht einfach: Dimension-Tabellen sollen so designt sein, dass Sie für alle Fact-Tabellen in allen Data Marts verwendet werden können:

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										

Abb. 4: DWH Dimensions Bus Architektur

Reporting

Nun sind die Daten also von der Staging Area in unsere als Star-Schema designten Data Marts geflossen und harren der Reports die da kommen mögen.

Um auf unseren Data Marts Reports zu erstellen bieten sich mehrere Möglichkeiten an: Wir können durch einfache SQL-Abfragen auf die Daten zugreifen. Diese Methode ist aber noch nicht sonderlich Manager tauglich. Wir können den interessierten Parteien aber auch `.CSV` Dateien zur Verfügung stellen, welche Sie dann in Ihrer Tabellenkalkulation nach Belieben nach den verschiedenen Dimensionen abfragen, sortieren und gruppieren können.

Ein direkter Zugriff über die Tabellenkalkulationssoftware auf die Datenbank ist ebenfalls via JDBC oder ODBC möglich.

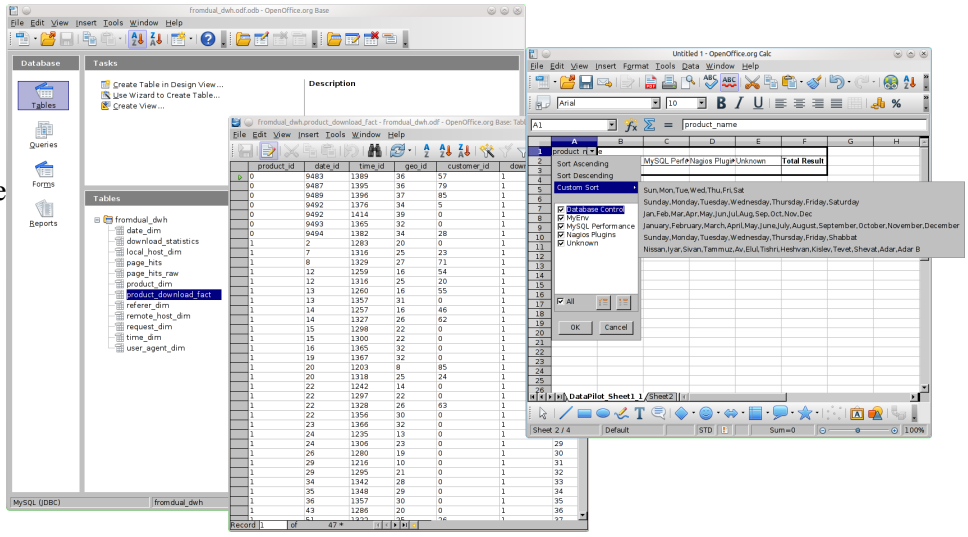

Abb. 5: Zugriff von der Tabellenkalkulation direkt ins Data Warehouse

Oder aber wir verwenden eine Reporting-Software, welche spezifisch auf die Wünsche der Datenempfänger zugeschnittene Reports auf Knopfdruck erstellt:

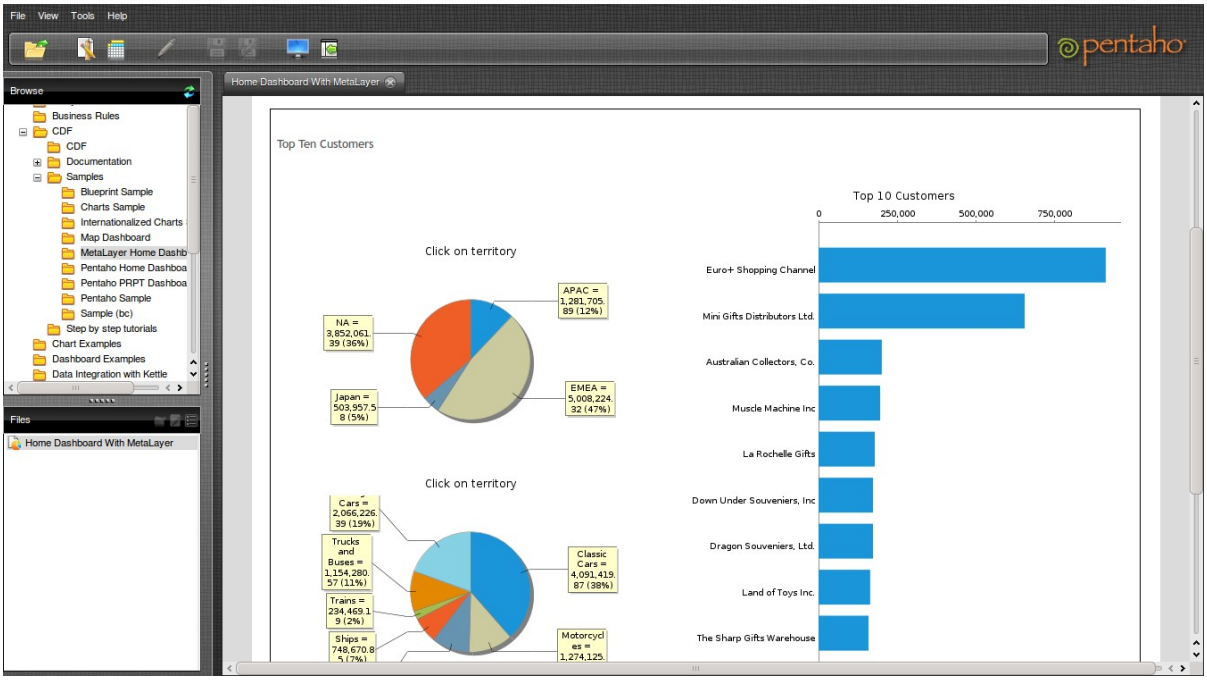

Abb. 6: Pentaho als Reporting-Software

Kontaktadresse:
Oli Sennhauser
FromDual GmbH
Rebenweg 6
CH – 8610 Uster

Telefon: +41 44 – 940 24 82
Fax: +41 43 – 55 68204
E-Mail oli.sennhauser@fromdual.com
Internet: www.fromdual.com