MySQL Conference - 2005

Building the Open Warehouse

Roger Magoulas
Market Research
O’Reilly Media
roger@oreilly.com
Introduction

- Market Research at O’Reilly
- Background: Data Warehouse Projects & Tools
  - Projects
    - San Francisco Opera
    - Sony Pictures
    - MyPlay (MP3 consolidator)
    - DigitalThink (on-line training)
    - O’Reilly
  - Tools
    - Oracle/SQL Server/DB2/MySQL
    - Epiphany/Informatica/Ardent
    - Brio/Cognos/Business Objects/Actuate/Crystal/Viador/SPSS
    - Alberta Auto Club
    - Sybase
    - QAD
    - Barter Site
    - Ofoto
Presentation Outline

- Why Open Source
- Architecture and Tools
  - Star Schema
  - ETL (Extraction/Transformation/Load)
  - Data Access
- Organizational Issues
Themes

- Mid-Sized Organizations
- DIY - Do It Yourself
- Fungible Technology Skills
- Practical Learnings and Style Notes

Terms

- Data Warehouse
- Data Mart
- ETL - Extraction/Transformation/Load
- Conditioning
- Star Schema/Dimensional Model
- ROLAP/MOLAP
Why Data Warehouse

**Answer Business Questions**
- Reporting Productivity and Consistency
- Analysis & Trends
- Understanding
- Decision Making
- Organizational Meme - Common Language
- Story Telling
  - makes data vivid
  - heuristic to remember what’s important
- Exploring the Cave of the Unknown
Open Source Data Warehouse If ...

- Can’t keep up w/ report/analysis requests
- No integration of data sources
  - No opportunity to consolidate system data
- OLTP data requires complex, slow SQL for reports
- Difficult to distribute reports
- Difficult to access data
- Fat clients difficult to maintain & require training
- Only system gurus understand source data
  - Understand anomalies & exceptions
- No opportunity to explore
Data Warehouse and Open Source

- **Data Warehouse as Perfect Open Source Project**
  - Less Stringent Operational Requirements
    - 24x7 Not Required
    - Reload to Recover
  - Gain Enterprise Open Source Experience
  - Data Warehouse Works on Open Source

- **Open Source great for Data Warehouse**
  - Reduces Cost
  - Use What You Know
  - Get Functionality You Need
  - Data Warehouse Works on Open Source
Data Warehouse Budget Considerations

- **Budget Reality for Mid-Sized Organizations**
  - $1 billion Revenue  => $1,000,000 budget
  - $100mm Revenue  => $ 100,000 budget

  - Little Room in Budget for Commercial Products Make Open Source Tools Critical for Data Warehouse

- **Yet, Greater Impact on Mid-Sized Organizations**
  - Cleaner Data
  - Fewer Sources to Integrate
  - Manageable Data Volumes

- **Focus on Business Considerations, Not Scaling**
Technical Framework

- **Hardware:**
  - SMP Server (consider 64 bit)
  - Serial ATA Disks / RAID 5 or 0+1
  - RAM++ (>4 Gb if 64 bit)

- **Software Platform - LAMP**
  - MySQL

- **Data Acquisition/ETL/Conditioning**
  - SQL, Perl

- **Data Access/Data Deliverable**
  - Web Delivery

- **Analysis:**
  - Excel
  - R (Real Statistics)
Data Warehouse Architecture

Sources
- Ops Systems
- External Sources
- 3rd Party, Web Logs & Flat Files
- Consolidated Source Copy

Warehouse System
- Consolidated Data Store
- ETL
- Data Movement
- Data Marts

Reporting
- Batch & Parametric Reports
- Graphical Navigation
- Power Users

Web Server
Why Dimensional Model

- Simpler queries
  - Fewer Tables, Simpler Joins
  - No Nulls or Outer Joins Required
  - Understandable Data
- Captures hierarchies for drilling up and down (OLAP)
- Conforming dimensions applicable to many topics/facts
- Easy to aggregate data for query performance
- Flexible and Extensible
  - Many Topics/Subjects
  - Conforming dimensions apply to multiple Fact Topics
  - Slowly Changing Dimensions Capture History
Dimensional Modeling - The Star Schema

- **Facts**
  - Normal
    - Keys and Measures
    - Thin and Long
  - Items to Count (or not): Measures
  - Grain
  - Factless Fact
  - Business, Science, Operational Subjects

- **Dimensions**
  - De-Normalized, i.e., Hierarchical
    - Wide and Short
    - Redundancy Encouraged, even Cardinal Products
  - Describes Characteristics of Fact
  - Conforming
  - Slowly Changing Dimensions
  - Degenerate Dimensions
Star Schema

- **calendar**
  - date_key
  - year
  - month
  - dom
  - format_date

- **product**
  - product_key
  - product_type
  - product_name
  - super_category
  - category
  - atomic_category
  - price

- **sales_fact**
  - date_key
  - product_key
  - author_key
  - customer_key
  - invoice_no
  - unit_sales

- **author**
  - author_key
  - author_name

- **customer**
  - customer_key
  - customer_name
  - location
  - sales_region
  - sales_territory
  - annual_rev
Star Schema - Snowflake

**calendar**
- date_key
- year
- month
- dom
- format_date

**product**
- product_key
- product_type
- product_name
- super_category
- category
- atomic_category
- price

**sales_fact**
- date_key
- product_key
- customer_key
- invoice_no
- unit_sales

**author_product**
- author_key
- author_name

**snowflake**
- author
- author_key
- author_name

**customer**
- customer_key
- customer_name
- location
- sales_region
- sales_territory
- annual_rev
Star Schema SQL

```sql
select
  c.year,
  sum(f.rev)  rev
from
  rev_fact    f
join calendar c
  on f.date_key = c.date_key
group by
  c.year
```

```
-------------------------
<table>
<thead>
<tr>
<th>year</th>
<th>rev</th>
</tr>
</thead>
</table>
-------------------------
| 2003 | 25250300|
-------------------------
```

```sql
select
  c.year,
  c.qtr,
  sum(f.rev)  rev
from
  rev_fact    f
join calendar c
  on f.date_key = c.date_key
group by
  c.year
  c.qtr
```

```
<table>
<thead>
<tr>
<th>year</th>
<th>qtr</th>
<th>rev</th>
</tr>
</thead>
<tbody>
<tr>
<td>2003</td>
<td>1</td>
<td>6100200</td>
</tr>
<tr>
<td>2003</td>
<td>2</td>
<td>5900400</td>
</tr>
<tr>
<td>2003</td>
<td>3</td>
<td>6620700</td>
</tr>
<tr>
<td>2003</td>
<td>4</td>
<td>6629000</td>
</tr>
</tbody>
</table>
```
Data Warehouse Architecture

Sources
- Ops Systems
- External Sources
- 3rd Party, Web Logs & Flat Files

Warehouse System
- Consolidated Data Store
- Consolidated Source Copy
- ETL (Data Movement)
- Data Marts

Reporting
- Batch & Parametric Reports
- Graphical Navigation
- Power Users

Web Server
Power Users
Ops Systems
External Sources
3rd Party, Web Logs & Flat Files
Consolidated Data Store
Consolidated Source Copy
ETL (Data Movement)
Data Marts
Web Server
CDS - Consolidated Data Store

- Near Copy of Relevant Source Data
  - Reduce Load on Source
  - Configure and Index for ETL
  - Tactical Reporting Source

- Importing Data From Source
  - Nightly Feeds
    - Source data stamp, triggers or check sum
  - Extraction Tools
    - Perl Wrapper or Shell Scripts
    - DBMS Fast Extraction Tools
      - SQL Server Issue
  - Parallel Extraction (and Load)
    - Shell Scripts
    - Dual Throttle Data Mover
  - Extract to Files and Load
    - Use Fast MySQL 'load data' command (myimport)
    - Big Data: Disable and Enable Index
  - XML
    - Makes Big Data Bigger
    - Adds Parse Step
ETL, Integration & Conditioning

- **Load Dimensional Model**
  - Integrate Sources
  - Clean Data
  - Transform:
    - Codes => Readable Form
    - Discrete Values => Ranges
  - Aggregate Data

- **Conditioning**
  - Perl Regex

- **ETL**
  - Query Logic => Data Model
  - **Intra Server SQL**
    - SQL-92 for efficiency
    - Transform: Lookup or Case
    - Perl Wrappers or Shell Script
  - **Procedural Logic Often Required**
SQL-92

select
  ...
from
  table_a a
  join table_b b
    on a.pkey = b.key
    and a.filter = 'value'
    and b.date_k > '2004-01-01'
  left join table_c c
    on b.bkey = c.bkey

- Finer query control
- Faster queries?
SQL Notes

- In table_a but not in table_b (filter in where)
  ```
  select
  ...
  from
  table_a a
  left join table_b b
  on a.pkey = b.pkey
  where
  b.pkey is null
  ```

- Delete and Insert Faster Than Update
  - Replace - requires primary key

- Manage Indexes at Load Time
  - Disable/Enable Indexes; load data local infile

- Subquery if clear, else 2-steps and stage table
  - 2-step allows index on ‘subquery’ results

- with Rollup - many levels results in single query

- Surrogate Key Option: prefix + source_key (hash)
  ```
  concat(‘a_’, source_key) as mart_key
  ```
Database Notes

- myisam files
  - No Logging
- Turn Query Cache on
- 5+ x Analysis Data for CDS/Mart Disk Capacity
  - RAID considerations
  - accommodate growth and batch jobs
- Index
  - Apply Carefully
    - primary keys of dimensions, foreign keys of fact
    - use explain
  - Separate indexes for compound Primary Key
- Memory/Heap Tables
  - Fact in memory
  - Use B-Tree
- Merge Tables
  - Partition Data
- Views - woo hoo

```
cREATE TABLE mheap (  
cust_key int,  
date_key date,  
amount int,  
index using btree(cust_key),  
index using btree(date_key)  
)  
type = memory
```
Data Warehouse Architecture

Sources
- Ops Systems
- External Sources
- 3rd Party, Web Logs & Flat Files

Warehouse System
- Consolidated Data Store
- ETL
- Data Movement

Reporting
- Batch & Parametric Reports
- Graphical Navigation
- Power Users

Data Marts
- Web Server

Consolidated Source Copy
Users & Analysis

- **User Distribution**
  - Power Users
  - Data Mavens
  - Data Manipulators
  - Report Readers
  - Data Pirates

- **Excel Matters**
  - Build Spreadsheets w/ Perl

- **Analysis**
  - Be Objective
  - Describe Concerns - Data Weakness
  - Simplify
  - Use Graphs - Tufte
  - Review Results & Interpret
  - Tell a Story
    - makes data vivid; memory heuristic
    - ‘Data is not information, information is not knowledge; knowledge is not wisdom’ Gary Flake, Yahoo Research
Data Access Tools

- **Canned and Parametric Reports**
  - **Perl**
    - Fast development; more flexible than GUI Tools
    - Hashes as multi-dimensional data structures
    - Pivot function
    - Integration w/ Excel
  - **Jasper**

- **Graphics, Analysis & Navigation**
  - **Java (Struts), Perl, PHP, Ruby on Rails, Flash, etc.**
  - **AJaX**
  - **R - real stats**
  - **Treemap/Dashboard**
  - **Tableau**

- **RSS**

- **Make Analysis Fun**
Query Template

- One SQL Query => Many Reports
- Perl Template Toolkit Example:

```perl
use template;
my $tt = Template->new();
my $vars;
my $sql;

$vars->{state_filter} = "'WA', 'OR', 'CA'";
$tt->process("*DATA, $vars, $sql");
print "$sql";

__DATA__
select
c.customer,
sum(f.rev)
from
customer c
join rev_fact f
  on c.ckey = f.ckey
  and state
    in ('WA', 'OR', 'CA')

DATA
```
Organizational Considerations

- **Medici Model**
- **Quantitative Culture**
  - MoneyBall
- **Build Fast & Iterate**
  - XP-like Quick, Tangible Results
- **Integrate Warehouse Build w/ Analysis**
  - Analysis creates warehouse data requirements
  - Supply-Side Analytics
- **Analytic environment requires constant attention**
  - Keep up w/ Business Changes, Systems Changes
- **Avoid Manual Dimension Assignments**
  - Natural Language Processing (NLP) and Categorization Algorithms (K-Mean, KNN, SVM) can help
    - but manual review required
    - initial training of data required
Other Topics

- Autofreq
- ‘The’ Report
  - Proxy for requirements
- Conventions
- Two Marts
  - as Backup
  - Limits downtime
- Save & Analyze Logs
- Statistical Feed QA
- Pivot Tables (the other pivot table)
- Date Dimension
- Full Text Search
- Set Constants
  - Help Optimizer
  - Relative Logic

```sql
select @sdate := date_sub(curdate(),
  interval (weekday(curdate())+8) day);

select ...
where date_key = @sdate
```
Performance

- Index Strategy
  - Careful Application
- Aggregate Data
  - Aggregate Navigation
- Key Cache
- Query Cache
- Analyze Tables
- Bulk Insert
- SQL-92
- Explain
- Slow Query Log
- RAID Striping
- Configuration
  - start w/ my-huge.cnf
References

- The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling - Ralph Kimball
- Visual Display of Quantitative Data - Edward Tufte
- MySQL - Paul DuBois
- MoneyBall - Michael Lewis
- Tipping Point - Malcolm Gladwell
- The Atlantic - Nation in Numbers
- Transact SQL Cookbook - Spetic & Gennick
- Template Toolkit - Chamberlain, Cross, Wardley
- High Performance MySQL - Jeremy Zawodny
- SQL For Smarties - Joe Celko
- Intelligent Enterprise
Open Source Data Warehouse Summary

- **Open Source**
  - Manage Costs
  - Functionality
- **CDS/Mart Architecture**
  - Consolidate Data
  - Star Schema
    - Make Data Understandable
  - Web Data Access & Delivery
  - Analyze & Iterate
- **Improve Reporting Productivity**
- **Improve Organizational Decision Making**
- **Flowers From Dirt**
## Crosstabs - Pivot

### Select Returns:

<table>
<thead>
<tr>
<th>Year</th>
<th>Platform</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004-01</td>
<td>java</td>
<td>1486</td>
</tr>
<tr>
<td>2004-02</td>
<td>java</td>
<td>2261</td>
</tr>
<tr>
<td>2004-03</td>
<td>java</td>
<td>1701</td>
</tr>
<tr>
<td>2004-04</td>
<td>java</td>
<td>1650</td>
</tr>
<tr>
<td>2004-05</td>
<td>java</td>
<td>2191</td>
</tr>
<tr>
<td>2004-06</td>
<td>java</td>
<td>2191</td>
</tr>
<tr>
<td>2004-01</td>
<td>linux</td>
<td>947</td>
</tr>
<tr>
<td>2004-02</td>
<td>linux</td>
<td>1191</td>
</tr>
<tr>
<td>2004-03</td>
<td>linux</td>
<td>925</td>
</tr>
<tr>
<td>2004-04</td>
<td>linux</td>
<td>868</td>
</tr>
<tr>
<td>2004-05</td>
<td>linux</td>
<td>1490</td>
</tr>
<tr>
<td>2004-06</td>
<td>linux</td>
<td>1329</td>
</tr>
</tbody>
</table>

### What You Want:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>java</td>
<td>1486</td>
<td>2261</td>
<td>1701</td>
<td>1650</td>
<td>2191</td>
<td>1564</td>
<td></td>
</tr>
<tr>
<td>linux</td>
<td>947</td>
<td>1191</td>
<td>925</td>
<td>868</td>
<td>1490</td>
<td>1329</td>
<td></td>
</tr>
</tbody>
</table>
Gratuitous 3-D Chart

- 80 Facings: 44
  - New Titles: 6
  - Digital Photo: 3
  - Other Apple: 8
  - Office: 4
  - iApp: 2
  - Mac OS X: 2

- 60 Facings: 33
  - New Titles: 5
  - Digital Photo: 6
  - Other Apple: 6
  - Office: 2
  - iApp: 3
  - Mac OS X: 4

- 40 Facings: 22
  - New Titles: 2
  - Digital Photo: 3
  - Other Apple: 2
  - Office: 1
  - iApp: 4
  - Mac OS X: 4

- 24 Facings: 13
  - New Titles: 2
  - Digital Photo: 6
  - Other Apple: 2
  - Office: 2
  - iApp: 4
  - Mac OS X: 4