

Migrating to MySQL

Jan Kneschke
Trainer & Consultant

MySQL GmbH

MySQLComCon 2004
Frankfurt/Main, Germany



About the Speaker

- **Jan Kneschke**
- Located in Kiel, Germany
- Maintainer of lighttpd, modlogan and pxtools
- Speaker at the PHPConf 2001, 2003 and 2004
- For PEAR: Image_GIS
- For MySQL: paradox Storage-Engine
- Trainer for .de, .at and .ch

What is Migration

1. to move from one country, place, or locality to another
2. to pass usually periodically from one region or climate to another for feeding or breeding
3. to change position in an organism or substance

Webster.com

- Level of Migration
 - Hardware
 - Database
 - SQL

Hardware Level

- Moving the same database to a new hardware
- MySQL to MySQL
- Make a backup
 - `mysqlhotbackup`
 - `mysqldump`
 - Shutdown + `tar czf datadir + my.cnf`
- MySQL binary updates are exchangeable between platforms, e.g.:
 - Windows to Unix
 - Intel to Sparc

Windows to Unix

- Windows uses backslashes, Unix slashes as path-seperators
- To keep it simple to migrate, path names should always be written with slashes in the configfiles
- Don't forget `SELECT INTO OUTFILE` and `LOAD DATA INFILE`
- Fake symlinks have to be replaced by real symlinks

Case Issues

- Directories and Tables are registered in the filesystem as directories and files
- NTFS can save the case in the filename, but can't distinguish between them
- FAT and all relatives always use UPPERCASE
- MacOS X HPFS is always lowercase
- All current Unix Filesystems are case sensitive
- Set `lower_case_table_names = 1`

Unix to Windows

- Please, don't
- Beside the case issue, it is 10-20% slower on the same hardware as on Linux
- Blocking reads
- ERROR 1067

Character Set

- Either use MySQL 4.1 or don't use non ASCII characters in tablename or database names:

```
CREATE TABLE `Streß test` (  
  `straße` CHAR(32) NOT NULL  
);
```

- If the servers don't use the same character set you will end up with broken table names
- MySQL 4.1 is using UTF-8 for everything in this scope

Transferring the Data

- There is no common way to move data transparently from one database to the other
- CSV is a last option
- ODBC bridges might be an option
 - ODBC might not provide to best mapping between the datatypes

Access to MySQL

- Use a CSV dump utility for Access
- Use mysqlimport or LOAD DATA INFILE to get the data into MySQL
- Use mdbtools from <http://mdbtools.sf.net/> to create a SQL dump
- Use the upcoming MDB table handler (more on this one in the talk at 15:15)
- Stay with Access as Frontend and use the ODBC to connect to MySQL
 - Don't forget that Access behaves differently in respect to ODBC from one to the next version
- PHP can read Access DB-files natively with the mdb-extension

Paradox to MySQL

- Paradox aka Borland Database Engine
- Use pxtools from <http://jan.kneschke.de/projects/pxtools/> to generate CSV or SQL dump
- The SQL dump can be generated for MySQL and other databases:
 - CREATE TABLE `tbl` ...
 - CREATE TABLE 'tbl' ...

Oracle to MySQL

- MySQL can't read Oracle's dump-format
- Use Oracles CSV export capability
 - MySQL has LOAD DATA INFILE and mysqlimport
- Use a handwritten Script which connects to both databases and is using SQL statements the fetch and insert the data.
- PHP and Perl can handle both databases nicely

SQL Level

- There are 3 SQL Standards
 - SQL92
 - SQL99
 - SQL:2003
- There are databases that are compliant to the standards
 - None of them has a big market-share
- Converting SQL itself has several pitfalls
- Don't talk about Stored Procedures

Abstraction

- Use a DB Abstraction layer
 - But which is the best level the DB should be abstracted at ?
 - Protocol Level: ODBC, JDBC, DBI
 - SQL Level: SQL-Parsers
 - Object Level: Hibernate, Propel for PHP
- The more abstraction you add the slower the data-access will get
- Abstraction removes the way for optimizing SQL queries completely
- Make your own decision based on your application

SQL - DDL

- Quoting literals: ticks vs. Backticks
- CREATE SCHEMA is CREATE DATABASE in MySQL
- Defaults for NULL
- Defining FOREIGN KEYS
- CHECK constrains

NULL default

- Input

```
CREATE TABLE tbl (  
    field CHAR(32)  
);
```
- MSSQL, Sybase

```
CREATE TABLE tbl (  
    field CHAR(32) NOT NULL  
);
```
- All other

```
CREATE TABLE tbl (  
    field CHAR(32) NULL  
);
```

VARCHAR ranges

- MySQL doesn't save whitespaces at the end of a VARCHAR
- CHAR don't return padding spaces at the end of the string
- Oracle has 4000 bytes max per VARCHAR, MySQL limits to 255 bytes
- MEDIUMTEXT, TEXT, LONGTEXT are a suitable replacement

VARCHAR2

- VARCHAR2 is quite often used to define storage for Unicode data
- MySQL uses the normal VARCHAR or TEXT types and adds a CHARACTER SET utf8 and a COLLATION if necessary

Integers

- Oracle uses NUMBER for everything
- MySQL has TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT with different ranges and storage-sizes
- DECIMAL in case of fixed point data

Reserved Words

- Don't use reserved words as field or tablename:
 - ORDER
 - DESC
 - VIEW
 - FROM
 - KEY
 - http://dev.mysql.com/doc/mysql/en/Reserved_words.html
- You can use them if you treat them as literals:

```
SELECT `KEY`  
FROM `ORDER`  
ORDER BY `DESC` DESC;
```

SQL - DML

- Quoting again
 - Only MySQL uses " and ' for quoting, all other just '
- Remove all kinds of Hints
 - SELECT STRAIGHT_JOIN ... FROM ... USE INDEX(...)
 - Strip it down the basic SQL
- CONCAT
 - CONCAT(col1, 'a', col2)
 - CONCAT(col1, CONCAT('a', col2))
 - col1 || 'a' || col2
 - col1 + 'a' + col2

JOIN or Subquery ?

- Oracle and some other databases prefer sub-selects
- MySQL prefers JOINS
- If in doubt use JOINS
- For rewriting subqueries to JOINS read
- Oracle prefers

`WHERE t1.col1 = t2.col2 (+)`

- All others use

`WHERE t1 LEFT JOIN t2 ON t1.col1 = t2.col2`

ROUND

- ROUND(3.5) might not do what you expect
 - MySQL on Windows: 3
 - MySQL on Unix: 4
- ROUND(2.5) looks even more confusing
 - MySQL on Windows: 2
 - MySQL on Unix: 2
- Use CEILING() and FLOOR() to define your own rounding rules

More NULL fun

- Oracle doesn't support the NULL
- An empty string represents the same as NULL in Oracle only

```
SELECT CONCAT('abc', NULL) FROM dual
```

Oracle: abc

all others: NULL

MySQL Gotchas

- Truncating STRINGS
- 1/0 is NULL, not an error
- If you reach the max possible value, the numbers don't increase and you don't get an error
- Implicate casts for

```
SELECT '2 apples' + '3 pears' + 4
```

- Generic Rule: MySQL will give you a warning, not an error.
- Second Rule: STRICT mode in MySQL 5.0 will fix this

VIEWS

- MySQL 4.1 doesn't support VIEWS
- Wait for 5.0 to get stable (Q1 2005)
- Use MaxDB
- Migrating VIEWS will be straight forward

Stored Procedures

- Every vendors has its own SP syntax
- They are not compatible to each other
- SQL:2003 has choosen the syntax of DB/2 as standard version
- MySQL 5.0 has implemented the SQL:2003 syntax
- The more SPs you use the harder to migration will be
- Do the neccesary in the DB, not all that is possible

Further Reading

- SQL Performance Tuning, Peter Gulutzan, Trudy Pelzer
- SQL99 Really Complete, Peter Gulutzan, Trudy Pelzer