

# Visualizing and Managing GIS-Data

A large, faint, light gray outline of the MySQL fish logo is positioned on the left side of the slide, partially overlapping the text area.

**Jan Kneschke**  
**Trainer & Consultant**

**MySQL GmbH**

**MySQLComCon 2004**  
**Frankfurt/Main, Germany**

## About the Speaker

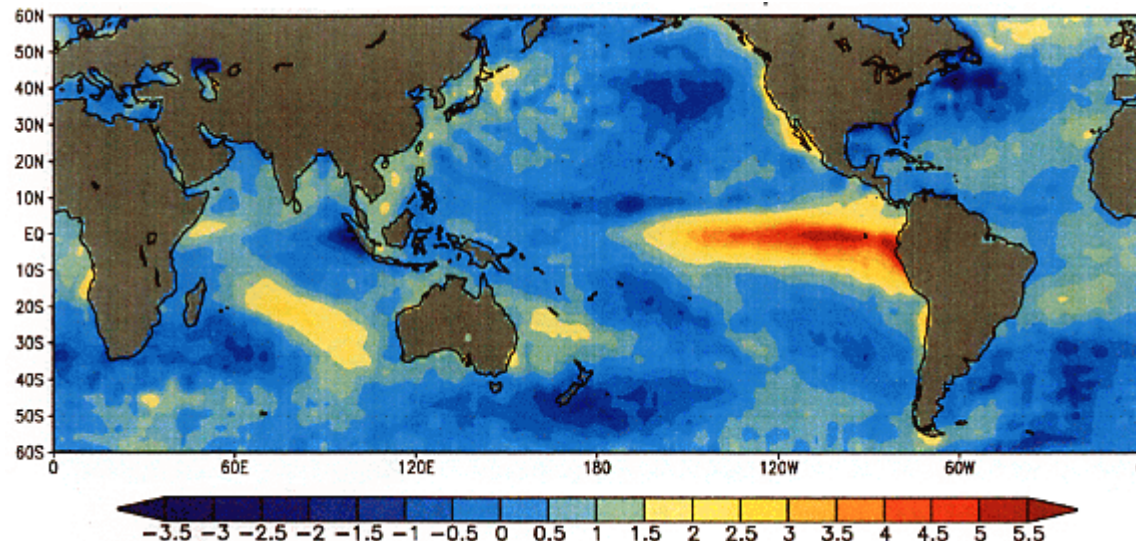
- **Jan Kneschke**
- Located in Kiel, Germany
- Maintainer of lighttpd, modlogan and pxtools
- Speaker at the PHPConf 2001, 2003 and 2004
- For PEAR: Image\_GIS
- For MySQL: paradox Storage-Engine
- Trainer for .de, .at and .ch

## Usage of GIS Data

- Base of scientific work in:
  - Geography
  - Geology
  - Sociology
- Routing, Mapping
- Geologic Analysis
- Environment Planning
- Demographics

## Usage of GIS Data

- Main purpose of GIS is combining geographic data with external datasets
- Remember the pictures of El-Niño



## GIS Support in MySQL 4.1

- GIS support included in MySQL 4.1
- Based on the OpenGIS specification
- Implements a subset of the full specification
  - Indexing of GIS datasets by using R-Trees
  - All datatypes available
  - Some functions missing

## Users of the GIS Support

- Citymap of Paderborn
  - High Resolution Map Generator for Users
  - Many requests
- Land Register of several counties in Northern Germany
  - Replication of common data
  - Every Land Register holds the data for the houses, streets and so on on the lowest level
  - Huge datasets

## GIS Data

- **Geographic Information System**
- Binding geographic objects to coordinates
- Simple Geometric Objects:
  - **Lines:** Streets, Rivers
  - **Polygons:** Political Boundaries
  - **Points:** Cities, Hills, Mountains
- Groups of minor objects
  - **Collections:** States, Countries, Continents

## Creating Datasets

```
CREATE TABLE geo (  
    geoobjid INT NOT NULL AUTO_INCREMENT,  
    line LINESTRING NOT NULL,  
    type ENUM('Border', 'Street', 'River',  
    'popPlace') NOT NULL;  
    name VARCHAR(32),  
    PRIMARY KEY (geoobjid),  
    SPATIAL KEY (line)  
);
```





## Importing Data - Datasources

- Shape files (SHP)
  - Binary format, easy to parse
  - Very popular in the US
  - Example project: <http://flightstats.us/>
  - Importer written by Jeremy Cole
  - libmygis
- ArcInfo (E00)
  - Base of the Digital Chart of the World (DCW)
  - Human Readable
  - Converters available in C, Perl and PHP

## Importing Data - ArcInfo

- Pairs of Points
- Building up connected Lines
- Grouped together to Polygons

```
1.4212879E+01 5.3864784E+01 1.4213489E+01 5.3884834E+01  
1.4212705E+01 5.3893837E+01 1.4208732E+01 5.3902344E+01  
1.4196489E+01 5.3910194E+01 1.4182546E+01 5.3914932E+01  
1.4206500E+01 5.3916485E+01 1.4224365E+01 5.3926537E+01
```

## Importing Data - INSERT

- Plain INSERT
- `GeomFromText()` to convert from WKT to internal format
- It's a bulk insert ! Use Multi-Row INSERTs

```
INSERT INTO de (line, type) VALUES  
(GeomFromText('LINESTRING(-69.2893  
-17.9578, -69.2963 -17.9504, ...)',  
'Border'));
```

## Selecting Data

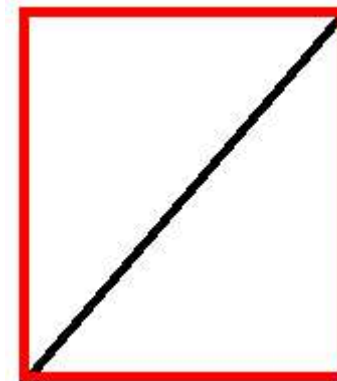
- It is SQL, use a SELECT
- AsText(), AsBinary()
  - Converting to WKT or WKB

```
SELECT AsText(line)
FROM geo
LIMIT 1;
```

```
LINestring(-70.3742 -18.3497, ...)
```

## Selecting Data - MBR

- **Minimum Bounding Rectangle**
- Every object has a MBR
- For a point it is a point
- For a horizontal or vertical line it is a line
- For everything else it is polygon with a area



```
SELECT AsText(Envelope(  
  GeomFromText('POINT(0 0)')));
```

```
POLYGON((0 0,0 0,0 0,0 0,0 0))
```

## Selecting Data - Zooming

- Take a rectangle range from your data
- Define a MBR around it
- SELECT the data

```
SELECT AsText(line)
FROM geo
WHERE MBRCONTAINS(
  GeomFromText(
    'LINESTRING(-69.2 -18.5, -69.4 -17)'),
  line);
```

## Selecting Data

- Data is only a set of LINESTRINGs
- Perfect for Drawing
- What is resulting MBR for all the Lines (or in this case the country) ?

```

SET @a = '';
SELECT @a := CONCAT(@a, IF(LENGTH(@a) > 0, ', ', '' ),
    AsText(line))
FROM geo
WHERE MBRContains(GeomFromText('LineString(-69.1 -18.5,
-70.9 -16)'), line);
SELECT AsText(Envelope(GeomFromText(CONCAT
('GeometryCollection(', @a, ')'))));

```



# Visualizing GIS Data

- Leaving the DB Area
- Theory
  - Projections
  - Coordinate Systems
  - Distance Calculations
- Practical Issues
  - Output Formats
  - Data Handling

# Projection

- Mapping a 3 dimensional object on 2 dimensions
- One of the following is loosing accuracy:
  - Distance (equidistant)
  - Bearing
  - Scale
  - Direction (azimuthal)
  - Shape(conformal)
  - Area (equiareal)
- Equator has about 40.000km, Northpole just a few kilometers, but still it is the same coordinate system when it comes to longitude and latitude

# Mapping

- Building a graphical projection
- Take a reference object
  - Cylindrical
  - Pseudocylindric
  - Conic
  - Pseudoconic
  - Azimuthal
- Choose what you want to preserve
  - Mercator
  - Azimuthal Equistant

# Mapping - Mercator



# Mapping - Azimuthal



## Back to Code

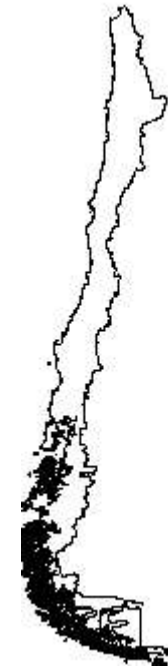
- PEAR::Image\_GIS
  - Parser for E00
  - Mapping for Mercator
  - Output for SVG and PNG
  - Very simple interface

```
<?php
require_once 'Image/GIS.php';
$map = new Image_GIS();

$map->addDataFile('ponet.e00', 'black');
$map->saveImage('somewhere.png');
?>
```

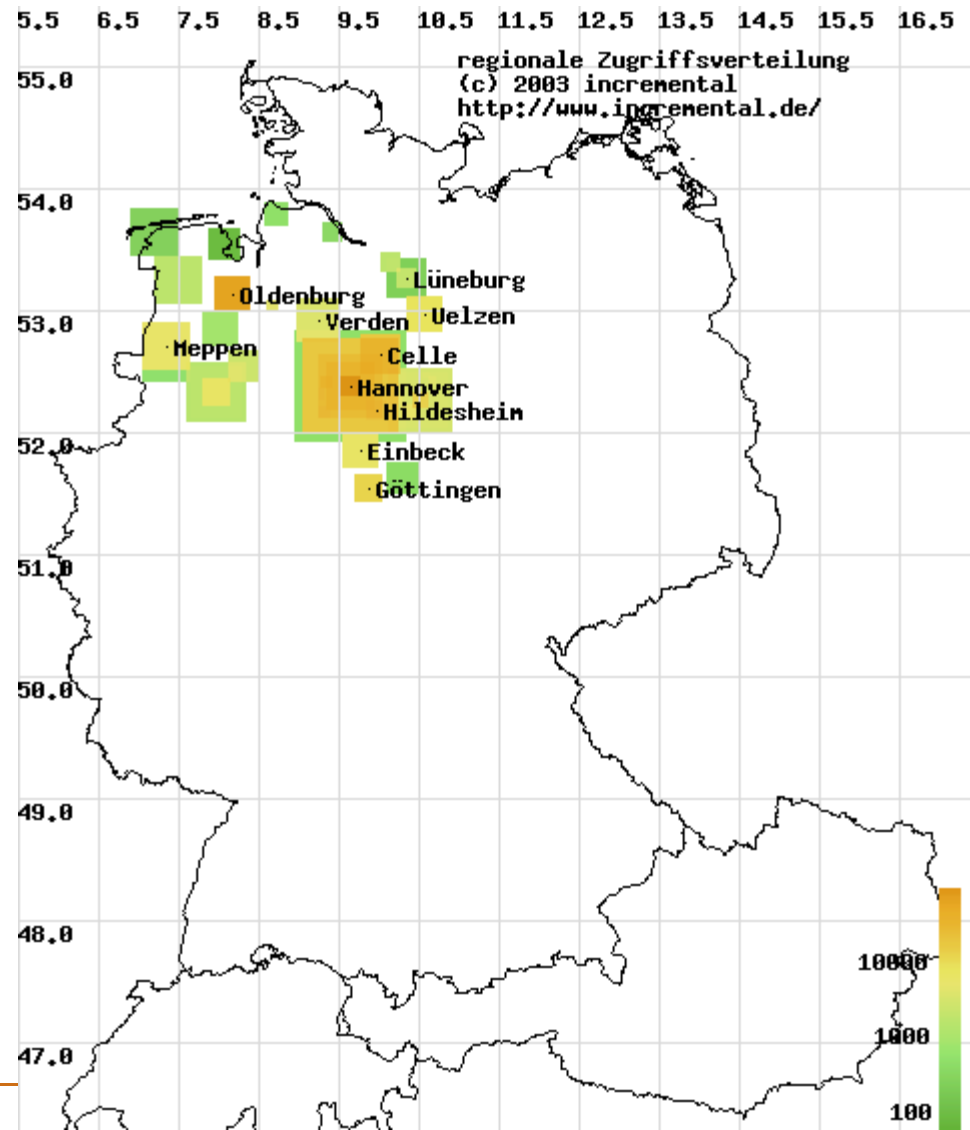
# Image\_GIS output

- South-America
  - Chile
  - A lot of cliffs at the south, pacific border
  - At the right bottom: border to fireland



# GIS Example

- Most visitors
- Lower Saxony
- Germany





## Further Reading

- GIS Support in MySQL: <http://www.mysql.com/gis>
- Rendering in PHP: [http://pear.php.net/image\\_gis](http://pear.php.net/image_gis)
- OpenGIS: <http://www.opengeospatial.org/>
- German Free GIS-data: <http://www.opengeodb.de/>
- Digital Chart of the World: <http://data.geocomm.com/>
- GEOnet Names Server  
<http://earth-info.nga.mil/gns/html/index.html>