Bochum, 9. November 2007

Quantitative Auswertungen der Wikipedia-Datenbank

Andreas Seier
as <at> aseier.de
http://www.aseier.de

Christoph Hassel
christoph.hassel <at> rub.de

Inhaltsverzeichnis

1 Ausgangsfrage

Die freie Internet-Enzyklopädie *Wikipedia* erfuhr in den vergangenen Jahren ein so beachtliches Wachstum, dass eine nähere sozialwissenschaftliche Untersuchung der Teilnahmemotivation und Interaktion weitreichende Erkenntnisse über das Verhalten von TeilnehmerInnen in „computervermittelten sozialen Netzwerken“ (Stegbauer2006: 90) zur kooperativen Inhaltserstellung verspricht. Diese Untersuchung soll quantitative empirische Daten für eine Diplomarbeit sammeln. Die Inhalte der Wikipedia stehen nicht nur zur Nutzung auf deren Internetseiten (http://wikipedia.org/) zur Verfügung, sondern werden zusätzlich in regelmäßigen Abständen als Database Dumps, also vollständige Sicherungen der Datenbank auf den Seiten der die Wikipedia tragenden Wikimedia Foundation zum Download angeboten (http://download.wikimedia.org/). Neben dem aktuellen Datenbestand umfasst das Angebot auch Sicherungen mit fast vollständiger Bearbeitungsgeschichte.[1] Erste Recherchen in Vorbereitung der geplanten Arbeit ergaben, dass dieser Umstand von der wissenschaftlichen Forschung bisher allenfalls in Ansätzen ausgeschöpft wurde.[2] Hieraus erwuchs die Absicht, den umfassenden Datenbestand eingehender zu untersuchen (vgl. auch Doering 2003: 235). Zu diesem Zweck wurde aus verschiedenen Gründen[3] die deutsche Wikipedia ausgewählt.

1.1 Ziele der Datengewinnung

Der vorliegende Datenbestand hat einerseits den großen Vorteil der Vollständigkeit. Es liegen alle Daten für alle in der gewählten Sprachversion jemals editierten Artikel vor und nicht lediglich eine Stichprobe. Andererseits stehen nur Daten und Metadaten zu den Artikeln und zugehörigen Diskussionen und ihrer Bearbeitungsgeschichte zur Verfügung. Eine Auswertung der Textinhalte ist mit derzeitiger Technik nur sehr eingeschränkt automatisiert möglich, da gleiche Sachverhalte unterschiedlich formuliert werden. So scheint Vandalismus an Artikeln bei Rückgängigmachung durchaus nicht immer mit diesem Begriff kenntlich gemacht zu werden. Daher fiel die Entscheidung, sich zunächst auf automatisiert auszuwertende Metadaten zu beschränken. Außerdem ist natürlich festzuhalten, dass nur

1 Mit Ausnahme einzelner, aus juristischen Gründen endgültig gelöschter Artikelversionen

2 Jakob Voss hat bereits 2005 eine Reihe quantitativer Auswertungen der Wikipedia-Datenbank vorgestellt (vgl. Voß 2005). Die Datengrundlage für diese Auswertungen wurde mit Hilfe der *Zachte2007* Perl-Scripts von Erik Zachte gesammelt (vgl. Zachte 2007). Auf der Website http://wm.sieheauch.de/, die den Autoren der vorliegenden Arbeit erst nach Abschluss der Datengewinnung bekannt geworden ist, werden die hiermit möglichen und weitere quantitative Auswertungen von ForscherInnen diskutiert.

3 hierzu zählt einerseits die Möglichkeit, mit AutorInnen der deutschen Wikipedia mündliche Leitfadeninterviews zu führen. Außerdem überstieg der Umfang der englischen Wikipedia in seiner Größe die Möglichkeiten der verwendeten Hardware. Da die deutsche Wikipedia nach Artikelzahl die zweitgrößte Sprachversion der freien Enzyklopädie darstellt und am 12. Mai 2001 nur wenige Monate nach der englischen Version gestartet wurde, konnte auch für die deutsche Ausgabe mit umfassenden und aussagekräftigen Daten gerechnet werden.

für den Betrieb der Wikipedia sinnvolle Metadaten erfasst und gespeichert werden. Einstellungen und Verhaltensweisen der BenutzerInnen können nur hieraus nur in Einzelfällen und auf spekulativer Grundlage erhoben werden, so dass sich ergänzende Befragungen von BenutzerInnen anbieten.

1.2 Relevante Daten

Nicht alle gewonnenen Daten werden in der Anschlussarbeit zur Auswertung herangezogen. Die folgende Liste der abgefragten Daten wurde primär anhand der sich aus der Datenstruktur ergebenden Möglichkeiten zusammengestellt. Die Umsetzung in SQL-Code wird weiter unten dargestellt. Die Abfragen teilen sich in eindimensionale Häufigkeitsverteilungen, zweidimensionale Häufigkeitsverteilungen und Listen.

1.2.1 Eindimensionale Häufigkeitsverteilungen

Die eindimensionalen Häufigkeitsverteilungen berücksichtigen den gesamten Bearbeitungszeitraum bis zum Stand der Daten.

- **Anzahl neuer Artikel je Zeitraum**
- **Gesamtzahl von Bearbeitungen je Zeitraum**
- **Anzahl anonymer Bearbeitungen je Zeitraum**
- **Neue BenutzerInnen je Zeitraum**
- **Anzahl aktiver BenutzerInnen** (solcher, die vor und nach oder im Zeitraum aktiv waren)
- **Anzahl je Zeitraum editierter Artikel**
- **Durchschnittliche Anzahl Bearbeitungen der im Zeitraum editierten Artikel**
- **Wachstum der Artikelzahl**
- **Revisionsanzahl je Zeitraum**
- **Wachstum der Revisionsanzahl**
- **Entwicklung der durchschnittlichen Textlänge eines Artikels**
- **Entwicklung der Gesamttextlänge**
- **Entwicklung der aktiven BenutzerInnen je Zeitraum**

- **Durchschnittliche Anzahl Bearbeitungen der im Zeitraum aktiven BenutzerInnen**
- **Entwicklung der BenutzerInnenzahl**
- **Entwicklung der durchschnittlichen Anzahl Bearbeitungen der BenutzerInnen bis Zeitpunkt**

1.2.2 Zweidimensionale Häufigkeitsverteilungen

- **Seitenbearbeitungen in Zeitraum**
- **Seitenbearbeitungen bis Zeitpunkt**
- **Artikellängen**
- **BenutzerInnenbearbeitungen in Zeitraum**
- **BenutzerInnenbearbeitungen bis Zeitpunkt**

1.2.3 Listen

- **Liste der Artikel mit den meisten bisherigen Bearbeitungen zu einem Zeitpunkt**
- **Liste der längsten Artikel zu einem Zeitpunkt**
- **Liste der Artikel mit den meisten Bearbeitungen innerhalb eines Zeitraumes**
- **Liste der BenutzerInnen mit den meisten Bearbeitungen bis zu einem Zeitpunkt**
- **Liste der BenutzerInnen mit den meisten Bearbeitungen innerhalb eines Zeitraumes**
- **Erster Artikel**

2 Beschaffung und Vorbereitung der Daten

Die Wikipedia verwendet das Mediawiki (http://www.mediawiki.org/) als Grundlage für die Erstellung und Präsentation ihrer Inhalte. Dieses wiederum setzt auf das Relationale Datenbank- Managementsystem MySQL (http://www.mysql.org/) auf, in der die Daten abgelegt werden. In der Vergangenheit wurde die Datenbank als sogenannter SQL-Dump[4]

4 SQL-Dumps sind das Standardverfahren von MySQL zur Sicherung von Datenbanktabelle. Sie enthalten die Tabellendaten, verpackt in SQL-Anweisungen, die die notwendigen Tabellen erstellen und mit dem Dateninhalt füllen.

zur Verfügung gestellt. U.a. aus technischen und aus Datenschutzgründen werden seit einiger Zeit jedoch nur noch XML-Dateien angeboten, die eigentlich zur Verarbeitung mit der frei erhältlichen Mediawiki-Software gedacht sind, die den Datenimport unmittelbar aus diesem Dateityp unterstützt (vgl. WM-Meta-DD, Abschnitt What_happened_to_the_SQL_dumps?).

2.1 Download der Daten

Zum Zeitpunkt des Downloads (08.08.2006) existierten nach Angaben auf der Downloadseite bereits neuere Dumps der deutschen Wikipedia. Die Gesamtdatenbank war jedoch in den jüngsten Dumps offensichtlich nicht oder nicht vollständig gesichert. Die Ursache hierfür war nicht ersichtlich. Nach Dateigröße urteilend lag die Vermutung nahe, dass die Datei vom 04.06.2006 (s. DE Wikipedia Database Dump, mittlerweile nicht mehr verfügbar) die letzte vollständige und erfolgreiche Sicherung enthalten müsste. Diese Vermutung konnte später nach einem Vergleich der Artikelzahl in der Sicherungsdatei und der von der Wikipedia angegebenen als mutmaßlich bestätigt eingestuft werden. Da die Dumps der vollständigen Editierhistorie im Kompressionsformat 7z besonders klein sind, wurde diese Variante bezogen (vgl. WM-Meta-DD).

2.2 Import der Daten

Als System kam Mac OS X 10.4 (zunächst auf einem iBook/PPC G4, 1GB RAM, später auf einem MacBook/Intel CoreDuo , 2GB RAM) zum Einsatz. Auf dem System wurde MySQL (Version 5.0.x; verwendete Version bei der Auswertung: 5.0.24, Download als Binärversion vom Hersteller: http://www.mysql.org/downloads/mysql/5.0.html#downloads) installiert. Zum Entpacken des Datenbankdumps wurde der Entpacker P7Zip (http://p7zip.sourceforge.net/, Version 4.42) im Quelltext heruntergeladen, übersetzt und installiert.

Für den Datenbankimport wurde zunächst ein Versuch unter Einsatz des Import-Features des Mediawiki unternommen, wofür der im System enthaltene Apache-Webserver um eine PHP-Installation ergänzt wurde (Download des Mediawiki in Version 1.5.2 von http://www.mediawiki.org/wiki/Download/). Ein erster Versuch zeigte sehr schnell, dass ein Import der Daten auf diese Weise nicht innerhalb eines überschaubaren Zeitraumes erfolgen würde. Außerdem erschien zweifelhaft, ob die vollständige Datenbank überhaupt entpackt auf der Festplatte unterzubringen wäre, so dass dieser Versuch abgebrochen wurde.

Alternativ bot sich die Verwendung des Java-Tools mwdumper (Download von http://download.wikimedia.org/tools/mwdumper.jar, Version vom 01.02.2006) an, das zu den von der Wikimedia Foundation gepflegten, zum Mediawiki gehörigen Hilfsprogrammen zählt

und dem Filtern und Konvertieren der XML-Dumps dient. (ebd., Abschnitt *Tools*). Mit mwdumper wurde der Datenbestand zunächst in einen SQL-Dump konvertiert[5]:

```
#!/bin/bash
7za e -so dewiki-20060604-pages-meta-history.xml.7z | \
  java -Xmx256M -jar mwdumper.jar --format=sql:1.5 | \
  gzip -2 > temp.sql.gz
```

Das Kommando schloss mit der Ausgabe

```
Everything is Ok
1.061.766 pages (17,161/sec), 16.133.524 revs (260,763/sec)
```

den Konvertierungsprozess erfolgreich ab. Eine kurze Einblicknahme in die erzeugte Ausgabe mit dem Befehl gzcat erweckte den Eindruck einer erfolgreichen Konvertierung.

Bevor Importversuche unternommen werden konnten, musste zunächst eine Datenbank mit entsprechenden Tabellen angelegt werden. In den zum Download angebotenen Daten sind nur die Tabellen mit den Artikel- und Revisionsdaten (konkret die Tabellen page, revision und text) enthalten. Bedauerlich, wenngleich aus der Perspektive des Datenschutzes nachvollziehbar, ist insbesondere, dass die user Tabelle nicht im Download enthalten ist. Die in ihr enthaltenen Informationen lassen sich aber teilweise – wie unten beschrieben – aus den Versionsgeschichten der Artikel rekonstruieren. Zum Anlegen der Datenbank wurde der MySQL-Kommandozeilenclient verwendet, der mit dem Befehl[6]

```
mysql -u root
```

aufgerufen wurde. Im MySQL-Client wurde die Datenbank mit dem Befehl

```
mysql> CREATE DATABASE wp2;
```

angelegt. Um den Import nicht weiter zu verkomplizieren, fiel die Entscheidung, die Tabellenstruktur der Originaldatenbank zunächst beizubehalten, auch wenn sie für die geplanten Abfragen nicht optimal war. Zum Anlegen der Tabellen konnte daher das Skript tables.sql aus der Mediawiki-Distribution verwendet werden. Das Anlegen der Tabellen erfolgte in der Shell mit dem Befehl

```
mysql -u root wp2 < tables.sql
```

Zur Durchführung des Imports und der unten beschriebenen Abfragen waren einige Änderungen der Konfiguration des MySQL-Servers erforderlich, so dass relativ große Datenblöcke verarbeitet werden konnten und große temporäre Speichertabellen möglich waren. Die durchgeführten Änderungen sind in der Konfigurationsdatei my.cnf vorzunehmen. Die Direktiven sind jeweils im richtigen Abschnitt der Konfigurationsdatei einzutragen:

5 In allen Codebeispielen wird davon ausgegangen, dass die verwendeten Programme im Pfad liegen und der Aufruf aus einem Verzeichnis data_mining_wikipedia erfolgt. Das Programm mwdumper liegt als jar-Datei vor und muss mit dem Befehl java aufgerufen werden, wozu es bei Verwendung des angegebenen Aufrufs ebenfalls im aktuellen Verzeichnis liegen sollte. Als Shell wurde die bash verwendet, was dem Systemstandard entspricht.

6 Die Verwendung von MySQL erfolgte für die Auswertung der Datenbank als Benutzer root. Dies ist auf einem Produktionssystem nicht zu empfehlen. Hinweise zur Benutzerverwaltung sind der Dokumentation zum MySQL-Server zu entnehmen (s. http://dev.mysql.com/doc/refman/5.0/en).

```
[mysqld]
port            = 3306
socket          = /tmp/mysql.sock
skip-locking
key_buffer = 1024M
max_allowed_packet = 32M
table_cache = 512M
sort_buffer_size = 8G
read_buffer_size = 1024M
read_rnd_buffer_size = 32M
myisam_sort_buffer_size = 1536M
thread_cache_size = 8M
query_cache_size= 512M
## options for optimization of sorts
max_heap_table_size = 1536M
myisam_max_sort_file_size = 8G
myisam_max_extra_sort_file_size = 8G

[myisamchk]
key_buffer = 8G
sort_buffer_size = 8G
read_buffer = 512M
write_buffer = 512M
```

Für eine sinnvolle Durchführung empfiehlt sich die Verwendung eines Systems mit relativ großem Arbeitsspeicher.

2.3 Reduzierung des Datenumfangs

Naheliegender Folgeschritt wäre der vollständige Import der Datenbank gewesen. Abgesehen von einer sehr geringen Ausführungsgeschwindigkeit bestanden abermals Befürchtungen, dass die vollständige Datenbank nicht entpackt auf der Festplatte Platz finden würde.

Aufgrund der bisherigen Erfahrungen mit der Verarbeitungsgeschwindigkeit für außerordentlich große Datenvolumina fiel nach kurzer Diskussion die Entscheidung, statt der Textinhalte aller Artikelversionen von vorn herein nur deren Länge (in ASCII-Zeichen) zu erfassen, da jede denkbare Auswertung der Artikelinhalte keine Aussichten auf eine akzeptable Ausführungsdauer gehabt hätte. Zu diesem Zweck wurde zunächst ein Versuch unternommen, die Artikellänge mit einem regulären Ausdruck (implementiert in der Scriptsprache Perl) zu erfassen, der jedoch nach mehreren Versuchen jeweils aufgrund besonders ungewöhnlicher Zeichenfolgen innerhalb von Artikeltexten abbrach. Nach unserer Schilderung des Problems erstellte Thomas Friedrichsmeier das C++ Programm *wikiconvert* eigens für den genannten Zweck, das sich unter Linux (Ubuntu 5.1) und MacOS (10.4) mit dem freien C++ Compiler aus der GNU Compiler Collection mit dem Aufruf

```
g++ wikiconvert.cpp -o wikiconvert
```

in ein ausführbares Binärprogramm übersetzen lässt. Im Verzeichnis data_-mining_wikipedia platziert erledigte es die Aufgabe klaglos mit dem Aufruf

```
cat temp.sql.gz | gunzip | ./wikiconvert | mysql -u root wp2
```

Insgesamt wurden so 37.615.422 Datensätze in drei Tabellen importiert.

3 Auswertung des Wikipedia-Datenbestandes nach dem Import in eine MySQL-Datenbank unter Verwendung von MySQL-Abfragen

3.1 Datenstruktur der nach dem Import vorliegenden MySQL-Datenbank

Nach dem zuvor beschriebenen Import des Datenbestands der deutschen Wikipedia lagen die Daten in einer MySQL-Datenbank (im weiteren Verlauf Komplettdatenbank oder auch Quelldatenbank genannt; MySQL-Name der Datenbank auf unserem System: wp2) mit folgender Tabellenstruktur vor:

```
mysql> use wp2;
Database changed
mysql> describe page;
+--------------------+----------------------+------+-----+---------+----------------+
| Field              | Type                 | Null | Key | Default | Extra          |
+--------------------+----------------------+------+-----+---------+----------------+
| page_id            | int(8) unsigned      | NO   | PRI | NULL    | auto_increment |
| page_namespace     | int(11)              | NO   | MUL |         |                |
| page_title         | varchar(255)         | NO   |     |         |                |
| page_restrictions  | tinyblob             | NO   |     |         |                |
| page_counter       | bigint(20) unsigned  | NO   |     | 0       |                |
| page_is_redirect   | tinyint(1) unsigned  | NO   |     | 0       |                |
| page_is_new        | tinyint(1) unsigned  | NO   |     | 0       |                |
| page_random        | double unsigned      | NO   | MUL |         |                |
| page_touched       | char(14)             | NO   |     |         |                |
| page_latest        | int(8) unsigned      | NO   |     |         |                |
| page_len           | int(8) unsigned      | NO   | MUL |         |                |
+--------------------+----------------------+------+-----+---------+----------------+
11 rows in set (0.33 sec)

mysql> describe revision;
+-----------------+----------------------+------+-----+---------+----------------+
| Field           | Type                 | Null | Key | Default | Extra          |
+-----------------+----------------------+------+-----+---------+----------------+
| rev_id          | int(8) unsigned      | NO   | PRI | NULL    | auto_increment |
| rev_page        | int(8) unsigned      | NO   | PRI |         |                |
| rev_text_id     | int(8) unsigned      | NO   |     |         |                |
| rev_comment     | tinyblob             | NO   |     |         |                |
| rev_user        | int(5) unsigned      | NO   | MUL | 0       |                |
| rev_user_text   | varchar(255)         | NO   | MUL |         |                |
| rev_timestamp   | char(14)             | NO   | MUL |         |                |
| rev_minor_edit  | tinyint(1) unsigned  | NO   |     | 0       |                |
| rev_deleted     | tinyint(1) unsigned  | NO   |     | 0       |                |
+-----------------+----------------------+------+-----+---------+----------------+
9 rows in set (0.08 sec)

mysql> describe text;
+-----------+------------------+------+-----+---------+----------------+
| Field     | Type             | Null | Key | Default | Extra          |
+-----------+------------------+------+-----+---------+----------------+
| old_id    | int(8) unsigned  | NO   | PRI | NULL    | auto_increment |
| old_text  | mediumblob       | NO   |     |         |                |
| old_flags | tinyblob         | NO   |     |         |                |
+-----------+------------------+------+-----+---------+----------------+
3 rows in set (0.07 sec)
```

3.2 Testdatenbank für die Entwicklung der MySQL-Abfragen

Da sich herausstellte, dass einzelne der im Folgenden beschriebenen Datenbank-Abfrage-Schritte für den gesamten importierten Datenbestand der deutschen Wikipedia wegen des großen Umfanges sehr lange dauerten (teilweise mehrere Tage), entschieden wir uns dafür, für die Entwicklung der Abfrage eine Testdatenbank (wp3) mit gleicher Datenbankstruktur zu erstellen, die sich auf 10.000 vom Zufallsgenerator ausgewählte Artikel beschränkte. Alle

folgenden MySQL-Abfragen wurden dann unter Verwendung dieser Testdatenbank (wp3) entwickelt und getestet. Aufgrund der Verwendung der gleichen Datenbankstruktur für die Testdatenbank (wp3) konnten die fertigen MySQL-Abfragen auch ohne Änderungen auf die Komplettdatenbank (wp2) angewendet werden.

3.2.1 Erstellen der Testdatenbank und Kopieren der ausgewählten Datensätze

Auch wenn die Testdatenbank nur für die Entwicklung der MySQL-Abfragen verwendet wurde und die daraus gewonnen Abfrageergebnisse nicht zur Grundlage der weiteren Auswertung verwendet wurden, sondern die Abfrageergebnisse der Komplettdatenbank, sei die Anlage der Testdatenbank an dieser Stelle kurz dokumentiert. Anlage und Verwendung einer Testdatenbank hat sich in unserem Fall als sehr hilfreich erwiesen und bei einer eventuellen Weiterentwicklung der hier beschriebenen MySQL-Abfragen empfehlen wir dieses Vorgehen, da es das Testen der entwickelten Abfragen ganz erheblich beschleunigt.

Das MySQL-Abfrage-Skript „Testdatenbank.sql“ findet sich unter: http://aseier.de/wikipedia/Skripte/Testdatenbank.sql.

Die Testdatenbank wurde, wie schon beschrieben, mit der gleichen Datenbankstruktur wie die Komplettdatenbank angelegt. Daher wird hier, was die Erläuterung der Datenbankstruktur und die Beschreibung des Erstellens der Testdatenbank und ihrer Tabellen angeht, auf den Abschnitt 2 verwiesen, in dem die Anlage der Komplettdatenbank beschrieben wird. In den „CREATE DATABASE“- und „CREATE TABLE“-Statements musste nur der Name der Datenbank von „wp2“ in „wp3“ geändert werden. Die für die Testdatenbank notwendigen CREATE-Statements sind in dem Skript „Testdatenbank.sql“ mit den beschriebenen Änderung enthalten und müssen nicht gesondert ausgeführt werden!

Ansonsten erläutern die Kommentare im Skript „Testdatenbank.sql“ das Kopieren der ausgewählten Artikel ausreichend. Als Zufallsgenerator bei der Auswahl der Artikel wurde die in MySQL eingebaute Funktion RAND() verwendet. Da wir die Abfrageergebnisse auf der Grundlage der Komplettdatenbank erstellt haben und die Testdatenbank nur zum Testen unserer MySQL-Abfragen benötigt haben, haben wir keine weitreichenden Überlegungen zur Repräsentativität der Datenauswahl in der Testdatenbank angestellt. Wollte man die im Weiteren beschriebene Auswertung der Wikipedia über die Testdatenbank durchführen, sei für die Bewertung der Ergebnisse folgender Auszug aus dem „MySQL 5.0 Referenz Manual“ (http://dev.mysql.com/doc/refman/5.0/en/mathematical-functions.html) nicht unerwähnt:

```
...
RAND() is not meant to be a perfect random generator,
but instead is a fast way to generate ad hoc random numbers ...
...
```

Das Skript „Testdatenbank.sql“ lässt sich mit folgendem Befehl ausführen:

```
mysql –u root < ./Skripte/Testdatenbank.sql > ./Ausgabe/Testdatebank.sql.out
```

Zuvor muss natürlich der zuvor in Abschnitt 2 beschriebene Import der Komplettdatenbank wp2 abgeschlossen sein. Nach der Ausführung des Skripts liegt die neu erstellte und gefüllte Testdatenbank wp3 vor. Achtung: Sollte vor der Ausführung des Skripts schon eine Datenbank mit dem Namen wp3 vorhanden sein, so wird diese durch das Skript ohne weitere Anmerkung oder Nachfrage gelöscht!

Die weitere Dokumentation unterscheidet auch nicht mehr zwischen einer Komplett- und einer Testdatenbank! Der weitere Text spricht von der „Quelldatenbank wp2", welche nach Befolgung der vorhergehenden Schritte der Komplettdatenbank entspricht. Für die Verwendung der Testdatenbank als Quelle muss in dem MySQL-Abfrage-Skript „Auswertung_Step_02.sql" die Skriptstellen „wp2." durch „wp3." ersetzt werden.

3.3 Die Auswertungs-Abfragen

Zur quantitativen Auswertung des wie oben beschrieben importierten Datenbestands der Wikipedia haben wir folgende sechs MySQL-Abfragen geschrieben, die mehrschrittig die importierten Daten auswerten und Informationen zu Statistischen Größen aggregieren.

1. Auswertung_Step_01.sql: tmp_time, tmp_class
2. Auswertung_Step_02.sql: tmp_page, tmp_rev, tmp_user
3. Auswertung_Step_03.sql: tmp_txp_in, tmp_txp_pre, tmp_txu_in, tmp_txu_pre
4. Auswertung_Step_04.sql: Ausgabe der Informationen ueber die Zeit
5. Auswertung_Step_05.sql: Ausgabe der Gesamt-Informationen
6. Auswertung_Step_06.sql: Ausgabe der Listen ueber die Zeit

Welche Aufgaben die einzelnen Abfragen einnehmen, entnehmen Sie bitten den Anmerkungen in den Abfragen.

Die Abfragen müssen in der aufsteigenden Reihenfolge ihrer Nummerierung durchgefuehrt werden, um korrekt zu funktionieren. Vor Ausführung dieser Abfragen muss natürlich wie in Abschnitt 2 beschrieben der Dump des Wikipedia-Datenbestands als mysql-Datenbank wp2 importiert worden sein. Wenn die Auswertung auf Grundlage der oben beschriebenen Testdatenbank erfolgen soll, ist die Abfrage „Auswertung_Step_02.sql" entsprechend zu verändern. Das Skript zur Ausführung aller Auswertungs-Abfrage sieht folgendermaßen aus:

```
#!/bin/bash

mysql -u root wp8 < ./Skripte/Auswertung_Step_01.sql > ./Ausgabe/Auswertung_Step_01.sql.out
mysql -u root wp8 < ./Skripte/Auswertung_Step_02.sql > ./Ausgabe/Auswertung_Step_02.sql.out
mysql -u root wp8 < ./Skripte/Auswertung_Step_03.sql > ./Ausgabe/Auswertung_Step_03.sql.out
mysql -u root wp8 < ./Skripte/Auswertung_Step_04.sql > ./Ausgabe/Auswertung_Step_04.sql.out
mysql -u root wp8 < ./Skripte/Auswertung_Step_05.sql > ./Ausgabe/Auswertung_Step_05.sql.out
mysql -u root wp8 < ./Skripte/Auswertung_Step_06.sql > ./Ausgabe/Auswertung_Step_06.sql.out
```

Nach erfolgreicher Ausführung aller sechs Abfragen liegen in den Dateien „Auswertung_Step_xx.sql.out“ die Ausgabeergebnisse der Abfragen bereit.

3.3.1 Die Auswertungs-Abfragen

Testdatenbank.sql

```
#
# MySQL 5.0.24
#

#
SELECT CURRENT_TIME();
SELECT 'Quantitative Auswertung der Wikipedia-Datenbank';
SELECT 'Abfrage Testdatenbank.sql';
#
# Dieses Abfrage legt die Testdatenbank wp3 an.
# Die Testdatenbank wp3 enthaelt 10.000
# mit dem Zufallsgenerator (mysql-Funktion RAND) ausgewaehlte Artikel
# der strukturgleichen kompletten Quelldatenbank wp2.
#
# Die Struktur der Quelldatenbank wird hier nicht noch einmal naeher erlaeutert.
#
#
# Weitere Erlaeuterungen zum Einsatz dieser mysql-Abfrage
# und zur Datenstruktur der Datenbanken:
#   "Quantitative Auswertung der Wikipedia-Datenbank"
#   von Christoph Hassel & Andreas Seier, 2006
#   http://aseier.de/wikipedia/
#


#
SELECT CURRENT_TIME();
SELECT 'Testdatenbank wp3 und ihre Tabellen (page, revision, text)';
#

#
SELECT CURRENT_TIME();
SELECT 'Entfernen einer eventuell schon existierenden Testdatenbank wp3';
#
DROP DATABASE IF EXISTS wp3;

#
SELECT CURRENT_TIME();
SELECT 'Erstellen der Testdatenbank wp3';
#
CREATE DATABASE wp3;

#
SELECT CURRENT_TIME();
SELECT 'Erstellen der Tabelle wp3.page';
#
CREATE TABLE wp3.page (
  page_id int(8) unsigned NOT NULL auto_increment,
  page_namespace int NOT NULL,
  page_title varchar(255) binary NOT NULL,
  page_restrictions tinyblob NOT NULL default '',
  page_counter bigint(20) unsigned NOT NULL default '0',
  page_is_redirect tinyint(1) unsigned NOT NULL default '0',
  page_is_new tinyint(1) unsigned NOT NULL default '0',
  page_random real unsigned NOT NULL,
  page_touched char(14) binary NOT NULL default '',
  page_latest int(8) unsigned NOT NULL,
  page_len int(8) unsigned NOT NULL,
  PRIMARY KEY page_id (page_id),
  UNIQUE INDEX name_title (page_namespace,page_title),
  INDEX (page_random),
  INDEX (page_len)
) TYPE=MyISAM;

#
SELECT CURRENT_TIME();
SELECT 'Erstellen der Tabelle wp3.revision';
#
CREATE TABLE wp3.revision (
  rev_id int(8) unsigned NOT NULL auto_increment,
  rev_page int(8) unsigned NOT NULL,
  rev_text_id int(8) unsigned NOT NULL,
  rev_comment tinyblob NOT NULL default '',
  rev_user int(5) unsigned NOT NULL default '0',
  rev_user_text varchar(255) binary NOT NULL default '',
  rev_timestamp char(14) binary NOT NULL default '',
```

```
  rev_minor_edit tinyint(1) unsigned NOT NULL default '0',
  rev_deleted tinyint(1) unsigned NOT NULL default '0',
  PRIMARY KEY rev_page_id (rev_page, rev_id),
  UNIQUE INDEX rev_id (rev_id),
  INDEX rev_timestamp (rev_timestamp),
  INDEX page_timestamp (rev_page,rev_timestamp),
  INDEX user_timestamp (rev_user,rev_timestamp),
  INDEX usertext_timestamp (rev_user_text,rev_timestamp)
) TYPE=MyISAM;

#
SELECT CURRENT_TIME();
SELECT 'Erstellen der Tabelle wp3.text';
#
CREATE TABLE wp3.text (
  old_id int(8) unsigned NOT NULL auto_increment,
  old_text mediumblob NOT NULL default '',
  old_flags tinyblob NOT NULL default '',
  PRIMARY KEY old_id (old_id)
) TYPE=MyISAM;


#
SELECT CURRENT_TIME();
SELECT 'Kopieren von ausgewaehlten Datensaetzen';
SELECT 'der Tabellen page, revision und text aus der Komplettdatenbank wp2';
SELECT 'in die entsprechenden Tabellen der Testdatenbank wp3';
#

#
SELECT CURRENT_TIME();
SELECT 'Kopieren von 10.000 durch den Zufallsgenerator (mysql-Funktion RAND)';
SELECT 'ausgewaehlten Datensaetzen der Tabelle wp2.page in die Tabelle wp3.page';
#

INSERT INTO wp3.page
  SELECT *
  FROM wp2.page
  ORDER BY RAND()
  LIMIT 10000         # Anzahl der Artikel in der Testdatenbank
;                     # Wert kann nach Bedarf geaendert werden!

#
SELECT CURRENT_TIME();
SELECT 'Kopieren der zu den ausgewaehlten und kopierten Datensaetze';
SELECT 'der Tabelle page gehoerigen Datensaetze der Tabelle wp2.revision';
SELECT 'in die Tabelle wp3.revision'
#

INSERT INTO wp3.revision
  SELECT wp2.revision.*
  FROM wp2.revision, wp3.page
  WHERE wp2.revision.rev_page = wp3.page.page_id
;

#
SELECT CURRENT_TIME();
SELECT 'Kopieren der zu den kopierten Datensaetze';
SELECT 'der Tabelle revision gehoerigen Datensaetze der Tabelle wp2.text';
SELECT 'in die Tabelle wp3.text'
#

INSERT INTO wp3.text
  SELECT wp2.text.*
  FROM wp2.text, wp3.revision
  WHERE wp2.text.old_id = wp3.revision.rev_text_id
;


#
SELECT CURRENT_TIME();
SELECT 'Ende der Abfrage!!!'
#
```

Auswertung_Step_01.sql

```
# MySQL 5.0.24

#
SELECT CURRENT_TIME();
SELECT 'Quantitative Auswertung der Wikipedia-Datenbank';
```

```
SELECT 'Abfrage Auswertung_Step_01.sql';
#
# Diese Abfrage ist die erste von sechs Abfragen, die die statistische Auswertung
# eines als mysql-Datenbank vorliegenden Wikipedia-Dumps ermoeglichen.
# Fuer die korreke Funkionsweise sollten die Abfragen in folgender Reihenfolge
# ausgefuehrt werden:
#
# 1. Auswertung_Step_01.sql # tmp_time, tmp_class
# 2. Auswertung_Step_02.sql # tmp_page, tmp_rev, tmp_user
# 3. Auswertung_Step_03.sql # tmp_txp_in, tmp_txp_pre, tmp_txu_in, tmp_txu_pre
# 4. Auswertung_Step_04.sql # Ausgabe der Informationen ueber die Zeit
# 5. Auswertung_Step_05.sql # Ausgabe der gesamt Informationen
# 6. Auswertung_Step_06.sql # Ausgabe der Listen
#
#
# Diese Abfrage bereitet die zeitliche und klassifizierende Auswertung
# der Wikipedia-Daten vor. Dazu werden mit Hilfe dieser Abfrage
# die beiden Tabellen tmp_time und tmp_class angelegt und mit einer unteren
# und einer oberen jeweils inkludierenden Intervall-Grenze versehen
# (Felder lim_min und lim_max).
#
# Die in dieser Abfrage angelegten Tabellen tmp_time_lim und tmp_class_lim
# bereiten die Erstellung der Tabellen tmp_time und tmp_class vor,
# indem darin die Grenzen definert werden und daraus fuer tmp_time und tmp_class
# zwei aufeinander folgende Grenzen als jeweils untere (lim_min) und
# obere (lim_max) Grenze der Zeit- bzw. Klassen-Intervalle herangezogen werden.
#
# Die in dieser Abfrage definierten Intervalle ermöglichen bzgl. der zeitlichen
# eine quartalsweise Auswertung und bzgl. der klassifizierenden Auswertung eine
# an der Zweierpotenzreihe orientierte Klassifizierung
# ( 0-0, 1-1, 2-2, 3-4, 5-8, 9-16, 17-32, ...).
#
#
# Weitere Erläuterungen zum Einsatz dieser mysql-Abfrage:
#   "Quantitative Auswertung der Wikipedia-Datenbank"
#   von Christoph Hassel & Andreas Seier, 2006
#   http://aseier.de/wikipedia/
#


#
SELECT CURRENT_TIME();
SELECT 'tmp_time_lim:';
SELECT 'Tabelle der Zeit-Grenzen (tmp_time_lim)';
#

#
SELECT CURRENT_TIME();
SELECT 'Entfernen einer eventuell schon existierenden Tabelle tmp_time_lim';
#
DROP TABLE
  IF EXISTS
  tmp_time_lim
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_time_lim:';
SELECT 'Erstellen der Tabelle tmp_time_lim';
#
CREATE TABLE
  IF NOT EXISTS
  tmp_time_lim (
    id  INT UNSIGNED  NOT NULL  AUTO_INCREMENT  KEY
  , lim BIGINT        NOT NULL                  UNIQUE  # Zeitgrenze
  )
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_time_lim:';
SELECT 'Füllen der Tabelle tmp_time_lim';
SELECT 'mit den Quartals-Grenzen von Jan 2000 bis Jan 2007';
#
INSERT INTO
  tmp_time_lim
  (lim)
VALUE
  ('20000100000000'),('20000400000000'),('20000700000000'),('20001000000000'),
  ('20010100000000'),('20010400000000'),('20010700000000'),('20011000000000'),
  ('20020100000000'),('20020400000000'),('20020700000000'),('20021000000000'),
```

```
  ('20030100000000'),('20030400000000'),('20030700000000'),('20031000000000'),
  ('20040100000000'),('20040400000000'),('20040700000000'),('20041000000000'),
  ('20050100000000'),('20050400000000'),('20050700000000'),('20051000000000'),
  ('20060100000000'),('20060400000000'),('20060700000000'),('20061000000000'),
  ('20070100000000')
;

#
# Für den Fall, dass die Auswertung nicht quartalsweise erfolgen soll,
# liegen hier die entsprechenden Zeit-Grenzen
# für eine jahresweise und monatsweise Auswertung vor.
# Das vorhergehende INSERT-Statement muss dazu
# durch eines der beiden folgenden INSERT-Statements ersetzt werden!
#

/*

#
SELECT CURRENT_TIME();
SELECT 'tmp_time_lim:';
SELECT 'Füllen der Tabelle tmp_time_lim';
SELECT 'mit den Jahres-Grenzen 2000 bis 2007';
#
INSERT INTO
  tmp_time_lim
  (lim)
VALUE
  ('20010000000000')
, ('20020000000000')
, ('20030000000000')
, ('20040000000000')
, ('20050000000000')
, ('20060000000000')
, ('20070000000000')


*/
/*

#
SELECT CURRENT_TIME();
SELECT 'tmp_time_lim:';
SELECT 'Füllen der Tabelle tmp_time_lim';
SELECT 'mit den Monats-Grenzen von Jan 2000 bis Jan 2007';
#
INSERT INTO
  tmp_time_lim
  (lim)
VALUE
  ('20000100000000'),('20000200000000'),('20000300000000'),
  ('20000400000000'),('20000500000000'),('20000600000000'),
  ('20000700000000'),('20000800000000'),('20000900000000'),
  ('20001000000000'),('20001100000000'),('20001200000000'),

  ('20010100000000'),('20010200000000'),('20010300000000'),
  ('20010400000000'),('20010500000000'),('20010600000000'),
  ('20010700000000'),('20010800000000'),('20010900000000'),
  ('20011000000000'),('20011100000000'),('20011200000000'),

  ('20020100000000'),('20020200000000'),('20020300000000'),
  ('20020400000000'),('20020500000000'),('20020600000000'),
  ('20020700000000'),('20020800000000'),('20020900000000'),
  ('20021000000000'),('20021100000000'),('20021200000000'),

  ('20030100000000'),('20030200000000'),('20030300000000'),
  ('20030400000000'),('20030500000000'),('20030600000000'),
  ('20030700000000'),('20030800000000'),('20030900000000'),
  ('20031000000000'),('20031100000000'),('20031200000000'),

  ('20040100000000'),('20040200000000'),('20040300000000'),
  ('20040400000000'),('20040500000000'),('20040600000000'),
  ('20040700000000'),('20040800000000'),('20040900000000'),
  ('20041000000000'),('20041100000000'),('20041200000000'),

  ('20050100000000'),('20050200000000'),('20050300000000'),
  ('20050400000000'),('20050500000000'),('20050600000000'),
  ('20050700000000'),('20050800000000'),('20050900000000'),
  ('20051000000000'),('20051100000000'),('20051200000000'),

  ('20060100000000'),('20060200000000'),('20060300000000'),
  ('20060400000000'),('20060500000000'),('20060600000000'),
```

```
  ('20060700000000'),('20060800000000'),('20060900000000'),
  ('20061000000000'),('20061100000000'),('20061200000000'),

  ('20070100000000')
;

*/

#
SELECT CURRENT_TIME();
SELECT 'tmp_time:';
SELECT 'Tabelle der Zeit-Intervalle (tmp_time)';
#

#
SELECT CURRENT_TIME();
SELECT 'tmp_time:';
SELECT 'Entfernen einer eventuell schon existierenden Tabelle tmp_time';
#
DROP TABLE
  IF EXISTS
  tmp_time
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_time:';
SELECT 'Erstellen der Tabelle tmp_time';
#
CREATE TABLE
  IF NOT EXISTS
  tmp_time (
    id      INT UNSIGNED  NOT NULL  AUTO_INCREMENT  KEY
  , lim_min BIGINT        NOT NULL                  UNIQUE
  , lim_max BIGINT        NOT NULL                  UNIQUE
  )
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_time:';
SELECT 'Füllen der Tabelle tmp_time mit den unteren und oberen Intervall-Grenzen';
#
# Zwei zuvor in der Tabelle tmp_time_lim definierte aufeinander folgenden Grenzen
# (time_lim_2.id = time_lim_1.id + 1) werden
# zur Festlegung der unteren (lim_min) und der oberen (lim_max) Grenze
# der Zeit-Intervalle herangezogen
#
INSERT INTO
  tmp_time(
    lim_min
  , lim_max
  )
SELECT
  time_lim_1.lim
, time_lim_2.lim
FROM
  tmp_time_lim AS time_lim_1
JOIN
  tmp_time_lim AS time_lim_2
ON
  time_lim_2.id = time_lim_1.id + 1
;


#
SELECT CURRENT_TIME();
SELECT 'tmp_class_lim:';
SELECT 'Tabelle der Klassen-Grenzen (tmp_class_lim)';
#

#
SELECT CURRENT_TIME();
SELECT 'tmp_class_lim:';
SELECT 'Entfernen einer eventuell schon existierenden Tabelle tmp_class_lim';
#
DROP TABLE
  IF EXISTS
  tmp_class_lim
;

#
SELECT CURRENT_TIME();
```

```
SELECT 'tmp_class_lim:';
SELECT 'Erstellen der Tabelle tmp_class_lim';
#
CREATE TABLE
  IF NOT EXISTS
  tmp_class_lim (
    id  INT UNSIGNED  NOT NULL  AUTO_INCREMENT  KEY
  , lim INT           NOT NULL                  UNIQUE
  )
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_class_lim:';
SELECT 'Füllen der Tabelle tmp_class_lim';
SELECT 'mit an der Zweierpotenzreihe orientierten Grenzen';
#
INSERT INTO
  tmp_class_lim
  ( lim         )
VALUE
  ( -1          )
, ( 0           )
, ( 1           )
, ( 2           )
, ( 4           )
, ( 8           )
, ( 16          )
, ( 32          )
, ( 64          )
, ( 128         )
, ( 256         )
, ( 512         )
, ( 1024        )
, ( 2048        )
, ( 4096        )
, ( 8192        )
, ( 16384       )
, ( 32768       )
, ( 65536       )
, ( 131072      )
, ( 262144      )
, ( 524288      )
, ( 1048576     )
, ( 2097152     )
, ( 4194304     )
, ( 8388608     )
, ( 16777216    )
;


#
SELECT CURRENT_TIME();
SELECT 'tmp_class:';
SELECT 'Tabelle der Klassen (tmp_class)';
#

#
SELECT CURRENT_TIME();
SELECT 'tmp_class:';
SELECT 'Entfernen einer eventuell schon existierenden Tabelle tmp_class';
#
DROP TABLE
  IF EXISTS
  tmp_class
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_class:';
SELECT 'Erstellen der Tabelle tmp_class';
#
CREATE TABLE
  IF NOT EXISTS
  tmp_class(
    id      INT UNSIGNED  NOT NULL  AUTO_INCREMENT  KEY
  , lim_min INT           NOT NULL                  UNIQUE
  , lim_max INT           NOT NULL                  UNIQUE
  )
;

#
```

```
SELECT CURRENT_TIME();
SELECT 'tmp_class:';
SELECT 'Füllen der Tabelle tmp_class mit den unteren und oberen Intervall-Grenzen';
#
# Zwei zuvor in der Tabelle tmp_class_lim definierte aufeinander folgenden Grenzen
# (class_lim_2.id = class_lim_1.id + 1) werden zur Festlegung
# der unteren (lim_min) und der oberen (lim_max) Grenze der Klassen-Intervalle
# herangezogen.
#
# Da im weiteren Verlauf der Abfragen der mysql-Vergleichsoperator
# "x BETWEEN lim_min AND lim_max" verwendet wird und dieser Vergleichsoperator
# äquivalent zu (lim_min <= x AND x <= lim_max) ist und damit
# die angegebenen Grenzen inkludiert werden, muss die untere Grenze
# eines Klassen-Intervalls um 1 gegenüber der oberen Grenze des vorhergegehenden
# Intervalls erhöht werden.
#
INSERT INTO
  tmp_class(
    lim_min
  , lim_max
  )
SELECT
  class_lim_1.lim + 1
, class_lim_2.lim
FROM
  tmp_class_lim AS class_lim_1
JOIN
  tmp_class_lim AS class_lim_2
ON
  class_lim_2.id = class_lim_1.id + 1
;


#
SELECT CURRENT_TIME();
SELECT 'Ende der Abfrage!!!'
#
```

Auswertung_Step_02.sql

```
# MySQL 5.0.24

#
SELECT CURRENT_TIME();
SELECT 'Quantitative Auswertung der Wikipedia-Datenbank';
SELECT 'Abfrage Auswertung_Step_02.sql';
#
# Diese Abfrage ist die zweite von sechs Abfragen, die die statistische Auswertung
# eines als mysql-Datenbank vorliegenden Wikipedia-Dumps ermoeglichen.
# Fuer die korreke Funkionsweise sollten die Abfragen in folgender Reihenfolge
# ausgefuehrt werden:
#
# 1. Auswertung_Step_01.sql # tmp_time, tmp_class
# 2. Auswertung_Step_02.sql # tmp_page, tmp_rev, tmp_user
# 3. Auswertung_Step_03.sql # tmp_txp_in, tmp_txp_pre, tmp_txu_in, tmp_txu_pre
# 4. Auswertung_Step_04.sql # Ausgabe der Informationen ueber die Zeit
# 5. Auswertung_Step_05.sql # Ausgabe der gesamt Informationen
# 6. Auswertung_Step_06.sql # Ausgabe der Listen
#
#
# In dieser Abfrage werden
#
# a) die auswertbaren Felder der Tabellen page, revision und text
# aus der Quelldatenbank wp2 in die neu erstellten Tabellen tmp_page und tmp_rev
# kopiert. Felder, die in der Quelldatenbank leer sind
# oder die fuer die Auswertung nicht benoetigt werden, werden weder
# in den Tabellen tmp_page angelegt, noch dorthin kopiert.
# Eventuell existierende Tabellen mit dem Namen tmp_page und tmp_rev werden
# zuvor geloescht! Felder werden mit dem Ziel einer systematischen Nomenklatur
# in unseren Abfragen teilweise gegenueber der Quelldatenbank anders benannt.
#
# b) die in wp2.revision, bzw. tmp_rev enthaltenen Informationen
# ueber registrierte Benutzer der Wikipedia fuer die Anlage der Tabelle tmp_user
# ausgewertet. Die eventuell schon existierende Tabelle tmp_user wird
# zuvor geloescht!
#
# c) aggregierte Informationen berechnet
#
# d) Informationen unter Anwendung der tmp_class und tmp_time klassifiziert
# bzw. Zeit-Intervallen zugeordnet. Zur Definition der Klassen und
```

```
# der Zeitgrenzen und Zeitintervalle in tmp_class und tmp_time siehe:
# Auswertung_Step_01.sql
#
# e) zahlreiche Indizes angelegt, um folgenden Abfragen ueber die Zieldatenbank
# zu beschleunigen.
#
# Da die vorliegenden Abfragen keine getrennte Auswertung der unterschiedlichen
# Namensraeume der Wikipedia (namespaces) vornimmt, muss in dieser Abfrage
# die Auswahl der Namensraeume vorgenommen werden.
#
# Die eigentlichen Artikel der Wikipedia sind im Seiten-Namensraum 0
# (page.namepace = 0) zu finden. Da fuer unser angestrebtes Ziel nur diese
# eigentlichen Artikel aus der Gesamtmenge der Seiten der Wikipedia benoetigt
# wurden und keine weiteren Beitraege der Wikipedia (Diskussionen,
# Benutzer-Selbstdarstellungen ...), wird beim Kopieren der Tabelle wp2.page
# in tmp_page nur diejenigen Datensaetze beruecksichtigt,
# die im Namensraum 0 liegen. Gleichwohl werden nur die Revisionen benoetigt,
# die den Artikel-Seiten (Namensraum 0) zugeordnet sind!
# Falls ein anderer Namensraum der Quelldatenbank ausgewertet werden soll,
# muss in dieser Abfrage die Angabe zum Namensraum geaendert werden
# (siehe dazu weiter unten).
#
# Die Quelldatenbank muss unter dem Namen wp2 angelegt sein, damit diese
# und somit auch die anschliessenden Abfragen funktionieren.
# Fuer den Fall, dass die Auswertung auf Grundlage der in den u.g. Erlaeuterung
# beschriebenen Testdatenbank geschehen soll, muessen in dieser Abfrage
# alle Stellen mit Angaben zur Quelldatenbank geaendert werden!!!
# ( wp2.  =>  wp3. )
#
# Als Zieldatenbank wird die in Benutzung befindliche existierende Datenbank
# verwendet! Zuvor schon in der Zieldatenbank angelegte Tabellen
# tmp_page, tmp_rev und tmp_user werden von dieser Abfrage geloescht!!!
#
#
# Wichtige Anmerkung
# zur Tabelle tmp_user
# und den Feldern user_anonym und user_id aus der Tabelle tmp_rev:
#
# In den uns vorliegendem Datenbestand der Wikipedia konnten wir Informationen
# ueber die registrierten Benutzer erst sinnvoll ab dem Quartal
# mit der time_id = 12 auswerten (also ab Okt. 2002).
# Dies resultiert daraus, dass im Dump der Wikipedia
# zum einen die Tabelle user nicht mit Inhalt geliefert wird und
# zum anderen bis einschliesslich zum Quartal mit der time_id = 11
# in der Tabelle revision die eindeutige Angabe rev_user
# nicht gesetzt ist. Die in tmp_user erfassten registrierten Benutzer
# ermitteln wir jedoch aufgrund der Nummer die in der Tabelle revision
# unter rev_user zu finden ist. Aussagen darueber, wieviele Revisionen
# bis zum Okt. 2002 durch registrierte Benutzer bzw. durch anonyme Benutzer
# erstellt wurden, lassen sich durch unsere Abfragen nicht machen. Ebenso kann
# nicht erfasst werden, ob Benutzer vor dem Quartal mit der time_id = 12 schon
# aktiv an Seiten der Wikipedia Revisionen vorgenommen haben.
# In der Tabelle revision liegen wohl durchgehend Angaben fuer das
# Feld rev_user_text vor, aus der sich Informationen zu den registrierten
# Benutzer bis zum Quartal mit der time_id = 11 ermitteln liessen.
# In dem besagten Feld wird er Name des registrierten Benutzers oder
# die IP-Adresse bzw. Namensumsetzung der IP-Adresse des anonymen Benutzers,
# der eine Revision erstellt hat, erfasst. Da aber auch Punkte in den Namen von
# registrierten Benutzern erlaubt sind, haben wir keinen befriedigenden Ausdruck
# gefunden, mit dem wir zuverlaessig feststellen konnten, ob es sich bei
# den erfassten Benutzern um registrierte oder anonyme handelt.
#
#
# Wichtige Anmerkung
# zu dem Feldern rev_first_id, rev_first_time_id, rev_last_id, rev_last_time_id
# der Tabellen tmp_page und tmp_user:
#
# Die Informationen zu den benannten Feldern der Tabelle tmp_user sind aufgrund
# der zuvor erwaehnten Anmerkungen zur Tabelle tmp_user in ihrer Aussage
# entsprechend vorsichtig zu bewerten. Erste und letzte Revisionen die
# bis zum einem bestimmten Zeitpunkt im Quartal mit der time_id = 11
# erstellt wurden, werden bei unseren Auswertungen nicht erfasst!
#
# Die Informationen zu den benannten Feldern der Tabelle tmp_page
# sind ebenfalls mit Vorsicht zu bewerten. Dies resultiert daraus, dass
# das Feld rev_id aus der Tabelle revision bis zum Quartal mit der time_id = 11
# einschliesslich nicht die gleiche Chronologie
# wie das Feld rev_timestamp aufweist. Da in dieser Abfrage die
# Felder rev_first_id und rev_last_id durch rev_first_id = MIN(tmp_rev.id)
```

```
# und rev_last_id = MAX(tmp_rev.id) ermittelt werden, sind Informationen
# der beiden Felder, bei denen rev_first_time_id und rev_last_time_id kleiner
# als 12 ist nicht zu verlaessig.
# Dieses Problem betrifft nicht
# die Felder rev_last_id, rev_last_len_id, rev_last_len der Tabelle tmp_txp_pre,
# da dort rev_last_id unter Zuhilenahme von rev_last_time_stamp ermittelt wird.
#
#
# Weitere Erlaeuterungen zum Einsatz dieser mysql-Abfrage:
#   "Quantitative Auswertung der Wikipedia-Datenbank"
#   von Christoph Hassel & Andreas Seier, 2006
#   http://aseier.de/wikipedia/
#


#
SELECT CURRENT_TIME();
SELECT 'tmp_page:';
SELECT 'Tabelle der Seiten (tmp_page)';
#
# Diese Tabelle enthaelt Informationen
# ueber die Seiten
# der Wikipedia.
#
# Folgende Felder werden aus der Tabelle page der Quelldatenbank wp2
# in die Tabelle tmp_page der Zieldatenbank uebernommen:
#
# - wp2.page.page_id -> tmp_page.id
# - wp2.page.page_namespace -> tmp_page.namespace_id
# - wp2.page.page_titel -> tmp_page.titel
#
# Neben diesen Feldern, enthaelt die Tabelle tmp_page
# folgende im Verlauf der Abfrage ermittelte Felder:
#
# - rev_count: Anzahl Revisionen der Seite,
#   d.h. wie haeufig die Seite geaendert wurde (Bearbeitungshaeufigkeit)
#
# - rev_count_id: Klassifizierung dieser Anzahl
#   (Klassen siehe Tabelle tmp_class in Auswertung_Step_01.sql)
#
# - rev_first_id: Nummer der ersten Revision der Seite,
#   d.h. Verweis auf die erste Reversion der Seite
#
# - rev_first_time_id: Nummer des Zeit-Intervalls, in dem die erste Revision liegt
#   d.h. in dem die Seite Seite angelegt wurde
#   (Zeitintervalle siehe Tabelle tmp_time)
#
# - rev_last_id: Nummer der letzten Revision der Seite
#   d.h. Verweis auf die letzte und damit zum Zeitpunkt des Datenbank-Dumps
#   aktuelle Revision der Seite
#
# - rev_last_len: Text-Laenge der letzten Revision
#
# - rev_last_len_id: Klassifizierung dieser Laenge
#   (Klassen siehe Tabelle tmp_class in Auswertung_Step_01.sql)
#
#
# Beachte auf jeden Fall die Anmerkungen
# zu dem Feldern rev_first_id, rev_first_time_id, rev_last_id, rev_last_time_id
# am Anfang dieser Abfrage!!!
#

#
SELECT CURRENT_TIME();
SELECT 'tmp_page:';
SELECT 'Entfernen einer eventuell schon existierenden Tabelle tmp_page';
#
DROP TABLE
  IF EXISTS
  tmp_page
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_page:';
SELECT 'Erstellen der Tabelle tmp_page';
#
CREATE TABLE
  IF NOT EXISTS
  tmp_page (
    id                INT UNSIGNED  NOT NULL  KEY
```

```
  , namespace_id      INT                         # Nummer des Namensraums
  , title             varchar(255)                # Titel der Seite

  , rev_count_id      INT                         # Klassifizierung der
                                                  #  Anzahl Revisionen
  , rev_count         INT                         # Anzahl Revisionen der Seite

  , rev_first_id      INT                         # Nummer der ersten Revision
  , rev_first_time_id INT                         # Nummer des Zeitintervalls
                                                  #  in der die erste Revision
                                                  #  erstellt wurde

  , rev_last_id       INT                         # Nummer der letzten Revision
  , rev_last_len_id   INT                         # Klassifizierung der Text-Laenge
                                                  #  der letzten Revision
  , rev_last_len      INT                         # Text-Laenge
                                                  #  der letzten Revision
  )
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_page:';
SELECT 'Kopieren der Datensaetze des Namensraums 0 bis 0';
SELECT '(wp2.page.page_namespace BETWEEN 0 AND 0)';
SELECT 'aus der Tabelle page der Quelldatenbank wp2,';
SELECT 'die nicht eine Umleitung auf eine andere Seite darstellen';
SELECT '(NOT wp2.page.page_is_redirect)';
SELECT 'in die Tabelle tmp_page der Zieldatenbank'
#
INSERT INTO
  tmp_page (
    id
  , namespace_id
  , title
  )
SELECT
  wp2.page.page_id
, wp2.page.page_namespace
, wp2.page.page_title
FROM
  wp2.page
WHERE
  NOT wp2.page.page_is_redirect                 # Keine Weitergeleitete Seiten
  AND wp2.page.page_namespace BETWEEN 0 AND 0   # HIER die Intervall der IDs
                                                # der Namespaces korrigieren,
                                                # die analysiert werden sollen!!!
;


#
SELECT CURRENT_TIME();
SELECT 'tmp_rev:';
SELECT 'Tabelle der Revisionen (tmp_rev)';
#
# Diese Tabelle enthaelt Informationen
# ueber die Revisionen
# der Seiten der Wikipedia.
#
# Folgende Felder werden aus der Tabelle revision der Quelldatenbank wp2
# in die Tabelle tmp_rev der Zieldatenbank uebernommen:
#
# - wp2.revision.rev_page -> tmp_rev.page_id
# - tmp_page.namespace_id -> tmp_rev.namespace_id
# - wp2.revision.rev_timestamp -> tmp_rev.time_stamp
# - NOT wp2.revision.rev_user -> tmp_rev.user_anonym
# - wp2.revision.rev_user -> tmp_rev.user_id
# - wp2.revision.rev_user_text -> tmp_rev.user_name
# - wp2.revision.rev_text_id -> tmp_rev.text_id
#
# Beachte auf jeden Fall die Anmerkungen
# zu dem Feldern user_anonym, user_id und user_text
# am Anfang dieser Abfrage!!!
#
# Neben diesen Feldern, enthaelt die Tabelle tmp_rev
# folgende im Verlauf der Abfrage ermittelte Felder:
#
# - time_id: Nummer des Zeit-Intervals, aus dem die Revision stammt
    (Zeit-Intervalle siehe Tabelle tmp_time in Auswertung_Step_01.sql)
# - text_len: Laenge des zugehoerigen Textes der Revision
# - text_len_id: Klassifizierung dieser Laenge
```

```
#   (Klassen siehe Tabelle tmp_class in Auswertung_Step_01.sql)

#
SELECT CURRENT_TIME();
SELECT 'tmp_rev:';
SELECT 'Entfernen einer eventuell schon existierenden Tabelle tmp_rev';
#
DROP TABLE
  IF EXISTS
  tmp_rev
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_rev:';
SELECT 'Erstellen der Tabelle tmp_rev';
#
CREATE TABLE
  IF NOT EXISTS
  tmp_rev (
    id            INT UNSIGNED  NOT NULL  KEY
  , page_id       INT           NOT NULL        # Nummer der zugehoerigen Seite
  , namespace_id  INT                           # Nummer des Namensraums
  , time_id       INT                           # Nummer des Zeit-Intervals
                                                #  indem Revision erstellt wurde
  , time_stamp    BIGINT                        # Zeit an dem Revision erstellt wurde
  , user_anonym   BOOL                          # Wird auf WAHR gesetzt, wenn
                                                #  user_id == 0
                                                #  (beachte Anmerkung weiter oben)
  , user_id       INT                           # Nummer des Benutzers, der die
                                                #  Revision verfasst hat
                                                #  (beachte Anmerkung weiter oben)
  , user_name     VARCHAR(255)                  # Name des Benutzer, bzw.
                                                #  IP oder Domain von dem / von der
                                                #  Revision verfasst wurde
                                                #  (beachte Anmerkung weiter oben)
  , text_id       INT                           # Nummer des zugehoerigen Textes
  , text_len_id   INT                           # Klassifizierung der Laenge
                                                #  des zugehoerigen Textes
  , text_len      INT                           # Laenge des zugehoerigen Textes
  )
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_rev:';
SELECT 'Kopieren der Datensaetze';
SELECT 'aus der Tabelle revision der Quelldatenbank wp2,';
SELECT 'in die Tabelle tmp_rev der Zieldatenbank,'
SELECT 'die den oben kopierten Seiten (siehe tmp_page) zugeordnet sind';
SELECT '(wp2.revision.rev_page = tmp_page.id)';
#
INSERT INTO
  tmp_rev (
    id
  , page_id
  , namespace_id
  , time_stamp
  , user_anonym
  , user_id
  , user_name
  , text_id
  )
SELECT
  wp2.revision.rev_id
, wp2.revision.rev_page
, tmp_page.namespace_id
, wp2.revision.rev_timestamp
, NOT wp2.revision.rev_user
, wp2.revision.rev_user
, wp2.revision.rev_user_text
, wp2.revision.rev_text_id
FROM (
  tmp_page
  JOIN (
    wp2.revision
  )
  ON
    wp2.revision.rev_page = tmp_page.id
)
;
CREATE INDEX  page_id       ON  tmp_rev (page_id);
```

```
CREATE INDEX  namespace_id  ON  tmp_rev (namespace_id);
CREATE INDEX  time_stamp    ON  tmp_rev (time_stamp);
CREATE INDEX  user_anonym   ON  tmp_rev (user_anonym);
CREATE INDEX  user_id       ON  tmp_rev (user_id);
CREATE INDEX  user_name     ON  tmp_rev (user_name);
CREATE INDEX  text_id       ON  tmp_rev (text_id);

#
SELECT CURRENT_TIME();
SELECT 'tmp_rev:';
SELECT 'Laenge des zugehoerigen Textes (text_len)';
SELECT 'aus der Tabelle text der Quelldatenbank wp2';
SELECT 'in die Tabelle tmp_rev der Zieldatenbank kopieren.';
#
UPDATE
  tmp_rev
, wp2.text
SET
  tmp_rev.text_len = wp2.text.old_text
WHERE
  wp2.text.old_id = tmp_rev.text_id
;
CREATE INDEX  text_len  ON  tmp_rev (text_len);

#
SELECT CURRENT_TIME();
SELECT 'tmp_page:';
SELECT 'Anzahl der Revisionen (rev_count) ';
SELECT 'und Nummer der ersten (rev_min_id) und letzten (rev_max_id) Revision';
SELECT 'ueber die Seiten ermitteln.';
#
REPLACE INTO
  tmp_page (
    id
  , namespace_id
  , title
  , rev_count
  , rev_first_id
  , rev_last_id
  )
SELECT
  tmp_page.id             # Feld wird nur auf sich selbst kopiert
, tmp_page.namespace_id   # Feld wird nur auf sich selbst kopiert
, tmp_page.title          # Feld wird nur auf sich selbst kopiert
, COUNT(tmp_rev.id)       # Anzahl der Revisionen
, MIN(tmp_rev.id)         # Nummer der ersten Revision
, MAX(tmp_rev.id)         # Nummer der letzten Revision
FROM (
  tmp_page
  JOIN (
    tmp_rev
  )
  ON
    tmp_rev.page_id = tmp_page.id
)
GROUP BY
  tmp_page.id
;
CREATE INDEX  namespace     ON  tmp_page (namespace_id);
CREATE INDEX  rev_count     ON  tmp_page (rev_count);
CREATE INDEX  rev_first_id  ON  tmp_page (rev_first_id );
CREATE INDEX  rev_last_id   ON  tmp_page (rev_last_id);

#
SELECT CURRENT_TIME();
SELECT 'tmp_rev:';
SELECT 'Nummer des Zeit-Intervalls (time_id),';
SELECT 'in dem der Zeitpunkt (time_stamp) der Erstellung der Revision liegt,';
SELECT '(Bearbeitungszeitpunkt) ueber die Revisionen ermitteln.';
#
UPDATE
  tmp_rev
, tmp_time
SET
  tmp_rev.time_id = tmp_time.id
WHERE
  tmp_rev.time_stamp BETWEEN tmp_time.lim_min AND tmp_time.lim_max
;
CREATE INDEX  time_id ON tmp_rev (time_id);

#
SELECT CURRENT_TIME();
SELECT 'tmp_page:';
```

```
SELECT 'Nummer des Zeit-Intervalls der ersten Revision (rev_first_time_id)';
SELECT 'ueber die Seiten ermitteln.';
#
UPDATE
  tmp_page
, tmp_rev
SET
  tmp_page.rev_first_time_id = tmp_rev.time_id
WHERE
  tmp_rev.id = tmp_page.rev_first_id
;
CREATE INDEX  rev_first_time_id ON tmp_page (rev_first_time_id);

#
SELECT CURRENT_TIME();
SELECT 'tmp_page:';
SELECT 'Klassifizierung der Anzahl der Revisionen';
SELECT 'ueber die Seiten ermitteln.';
#
UPDATE
  tmp_page
, tmp_class
SET
  tmp_page.rev_count_id = tmp_class.id
WHERE
  tmp_page.rev_count BETWEEN tmp_class.lim_min AND tmp_class.lim_max
;
CREATE INDEX  rev_count_id ON  tmp_page (rev_count_id);

#
SELECT CURRENT_TIME();
SELECT 'tmp_rev:';
SELECT 'Klassifizierung der Text-Laenge';
SELECT 'ueber die Revisionen ermitteln.';
#
UPDATE
  tmp_rev
, tmp_class
SET
  tmp_rev.text_len_id = tmp_class.id
WHERE
  tmp_rev.text_len BETWEEN tmp_class.lim_min AND tmp_class.lim_max
;
CREATE INDEX  text_len_id ON  tmp_rev (text_len_id);

#
SELECT CURRENT_TIME();
SELECT 'tmp_page:';
SELECT 'Texte-Laenge und Klassifizierung der Text-Laenge';
SELECT 'der letzten zugehoerigen, d.h. aktuellen Revision';
SELECT 'ueber die Seiten kopieren.';
#
UPDATE
  tmp_page
, tmp_rev
SET
  tmp_page.rev_last_len = tmp_rev.text_len
, tmp_page.rev_last_len_id = tmp_rev.text_len_id
WHERE
  tmp_rev.id = tmp_page.rev_last_id
;
CREATE INDEX  rev_last_len_id ON  tmp_page (rev_last_len_id);


#
SELECT CURRENT_TIME();
SELECT 'tmp_user:';
SELECT 'Tabelle der Benutzer (tmp_user)';
#
# In der Tabelle der Benutzer (tmp_user) werden Informationen ueber die
# eingetragenen Benutzer gesammelt. Da der Dump der Wikipedia die Tabelle der
# Benutzer nicht zur Verfuegung stellt, blieb nur die Moeglichkeit,
# Informationen ueber die eingetragenen Benutzer aus der Tabelle der Revisionen
# (wp2.revision bzw. tmp_rev) zu aggregieren.
#
# Beachte auf jeden Fall die Anmerkungen
# zur Tabelle tmp_user und
# den Feldern rev_first_id, rev_first_time_id, rev_last_id, rev_last_time_id
# am Anfang dieser Abfrage!!!
#

#
SELECT CURRENT_TIME();
```

```
SELECT 'tmp_user:';
SELECT 'Entfernen einer eventuell schon existierenden Tabelle tmp_page';
#
DROP TABLE
  IF EXISTS
  tmp_user
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_user:';
SELECT 'Erstellen der Tabelle tmp_user';
#
CREATE TABLE
  IF NOT EXISTS
  tmp_user (
    id                INT UNSIGNED  NOT NULL KEY
  , name              VARCHAR(255)  NOT NULL      # Name des Benutzers
  , rev_count_id      INT                         # Klassifizierung
                                                  #  Anzahl der Revisionen
  , rev_count         INT                         # Anzahl der Revisionen, die
                                                  #  vom Benutzer erstellt wurden
  , rev_first_id      INT                         # Nummer der ersten Revision,
                                                  #  die der Benutzer erstellt hat
  , rev_first_time_id INT                         # Nummer des Zeitintervalls,
                                                  #  in dem der Benutzer seine
                                                  #  erste Revision erstellt hat
  , rev_last_id       INT                         # Nummer der letzten Revision,
                                                  #  die der Benutzer erstellt hat
  , rev_last_time_id  INT                         # Nummer des Zeitintervalls,
                                                  #  in dem der Benutzer seine
                                                  #  letzte Revision erstellt hat
  )
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_user:';
SELECT 'Ermittle Nummer und Name der Benutzer, die im ausgewaehlten Namensraum';
SELECT 'Revisionen erstellt haben, aus der Tabelle tmp_rev.';
SELECT 'Ermittle die Anzahl, die Erste und die Letzte der Revisionen,';
SELECT 'die der Benutzer im ausgewaehlten Namensraum erstellt hat.';
#

INSERT INTO
  tmp_user (
    id
  , name
  , rev_count
  , rev_first_id
  , rev_last_id
  )
SELECT
  user_id
, user_name
, COUNT(id)         # Anzahl der Revisionen
, MIN(id)           # Nummer der ersten Revision
, MAX(id)           # Nummer der letzten Revision
FROM
  tmp_rev
WHERE
  user_id           # nur Revisionen von eingetragenen Benutzern ermitteln
GROUP BY
  user_id
;
CREATE INDEX  name          ON  tmp_user (name);
CREATE INDEX  rev_count     ON  tmp_user (rev_count);
CREATE INDEX  rev_first_id  ON  tmp_user (rev_first_id);
CREATE INDEX  rev_last_id   ON  tmp_user (rev_last_id);


#
SELECT CURRENT_TIME();
SELECT 'tmp_user:';
SELECT 'Klassifizierung der Anzahl der Revisionen';
SELECT 'ueber die Benutzer ermitteln.';
#
UPDATE
  tmp_user
, tmp_class
SET
  tmp_user.rev_count_id = tmp_class.id
WHERE
```

```
  tmp_user.rev_count BETWEEN tmp_class.lim_min AND tmp_class.lim_max
;
CREATE INDEX  rev_count_id ON  tmp_user (rev_count_id);


#
SELECT CURRENT_TIME();
SELECT 'tmp_user:';
SELECT 'Nummer des Zeit-Intervalls der ersten Revision (rev_first_time_id)';
SELECT 'ueber die Benutzer ermitteln.';
#
UPDATE
  tmp_user
, tmp_rev
SET
  tmp_user.rev_first_time_id = tmp_rev.time_id
WHERE
  tmp_rev.id = tmp_user.rev_first_id
;
CREATE INDEX  rev_first_time_id ON tmp_user (rev_first_time_id);


#
SELECT CURRENT_TIME();
SELECT 'tmp_user:';
SELECT 'Nummer des Zeit-Intervalls der letzten Revision (rev_last_time_id)';
SELECT 'ueber die Benutzer ermitteln.';
#
UPDATE
  tmp_user
, tmp_rev
SET
  tmp_user.rev_last_time_id = tmp_rev.time_id
WHERE
  tmp_rev.id = tmp_user.rev_last_id
;
CREATE INDEX  rev_last_time_id ON tmp_user (rev_last_time_id);


#
SELECT CURRENT_TIME();
SELECT 'Ende der Abfrage!!!'
#
```

Auswertung_Step_03.sql

```
# MySQL 5.0.24

#
SELECT CURRENT_TIME();
SELECT 'Quantitative Auswertung der Wikipedia-Datenbank';
SELECT 'Abfrage Auswertung_Step_03.sql';
#
# Diese Abfrage ist die dritte von sechs Abfragen, die die statistische Auswertung
# eines als mysql-Datenbank vorliegenden Wikipedia-Dumps ermoeglichen.
# Fuer die korreke Funkionsweise sollten die Abfragen in folgender Reihenfolge
# ausgefuehrt werden:
#
# 1. Auswertung_Step_01.sql # tmp_time, tmp_class
# 2. Auswertung_Step_02.sql # tmp_page, tmp_rev, tmp_user
# 3. Auswertung_Step_03.sql # tmp_txp_in, tmp_txp_pre, tmp_txu_in, tmp_txu_pre
# 4. Auswertung_Step_04.sql # Ausgabe der Informationen ueber die Zeit
# 5. Auswertung_Step_05.sql # Ausgabe der gesamt Informationen
# 6. Auswertung_Step_06.sql # Ausgabe der Listen
#
#
# In dieser Abfrage vier Kreuztabellen angelegt, die die Auswertung
# der Wikipedia-Daten ueber die Zeit ermoeglichen.
# Im einzelnen handelt es sich um folgende Tabellen:
#
# - tmp_txp_in: Informationen ueber die Seiten innerhalb eines Zeitintervalls
# - tmp_txp_pre: Informationen ueber die Seiten vor einem Zeitpunkt
# - tmp_txu_in: Informationen ueber die Benutzer innerhalb eines Zeitintervalls
# - tmp_txu_pre: Informationen ueber die Benutzer vor einem Zeitpunkt
#
# Zur Definition der Zeitgrenzen und der Zeitintervalle in tmp_time siehe:
# Auswertung_Step_01.sql
#
#
# Weitere Erlaeuterungen zum Einsatz dieser mysql-Abfrage:
#   "Quantitative Auswertung der Wikipedia-Datenbank"
#   von Christoph Hassel & Andreas Seier, 2006
```

```
#   http://aseier.de/wikipedia/
#


#
SELECT CURRENT_TIME();
SELECT 'Kreuztabelle der Seiten ueber die Zeit (tmp_txp_in)';
SELECT 'zur Auswertung innerhalb der Zeitintervalle';
#

#
SELECT CURRENT_TIME();
SELECT 'tmp_txp_in:';
SELECT 'Entfernen einer eventuell schon existierenden Tabelle tmp_txp_in';
#
DROP TABLE
  IF EXISTS
  tmp_txp_in
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_txp_in:';
SELECT 'Erstellen der Tabelle tmp_txp_in';
#
CREATE TABLE
  IF NOT EXISTS
  tmp_txp_in (
    id                INT UNSIGNED  NOT NULL AUTO_INCREMENT  KEY
  , time_id           INT # Nummer des Zeitintervalls
  , page_id           INT # Nummer der Seite
  , namespace_id      INT # Nummer des Namensraums der Seite
  , rev_count_in_id   INT # Klassifizierung der Anzahl der Revisionen
  , rev_count_in      INT # Anzahl der Revisionen der Seite innerhalb eines Zeitintervalls
  )
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_txp_in:';
SELECT 'Anzahl der Revisionen innerhalb eines Zeitintervalls (rev_count_in)';
SELECT 'ueber die Seiten und die Zeit ermitteln.';
#
INSERT INTO
  tmp_txp_in (
    time_id
  , page_id
  , namespace_id
  , rev_count_in
  )
SELECT
  tmp_time.id
, tmp_page.id
, tmp_page.namespace_id
, COUNT(tmp_rev.id)
FROM (
  tmp_time
  JOIN (
    tmp_page
    JOIN
      tmp_rev
    ON
      tmp_rev.page_id = tmp_page.id
  )
  ON
    tmp_rev.time_id = tmp_time.id
)
GROUP BY
  tmp_time.id
, tmp_page.id
;
CREATE UNIQUE INDEX txp_id ON tmp_txp_in (time_id, page_id);
CREATE INDEX time_id ON tmp_txp_in (time_id);
CREATE INDEX page_id ON tmp_txp_in (page_id);
CREATE INDEX namespace_id ON tmp_txp_in (namespace_id);
CREATE INDEX rev_count_in ON tmp_txp_in (rev_count_in);


#
SELECT CURRENT_TIME();
SELECT 'tmp_txp_in:';
SELECT 'Klassifizierung der Anzahl der Revisionen (rev_count_in_id)';
```

```
#
UPDATE
  tmp_txp_in
, tmp_class
SET
  tmp_txp_in.rev_count_in_id = tmp_class.id
WHERE
  tmp_txp_in.rev_count_in BETWEEN tmp_class.lim_min AND tmp_class.lim_max
;
CREATE INDEX  rev_count_in_id ON  tmp_txp_in (rev_count_in_id);


#
SELECT CURRENT_TIME();
SELECT 'tmp_txp_pre:';
SELECT 'Kreuztabelle der Seiten ueber die Zeit (tmp_txp_pre)';
SELECT 'zur Auswertung vor einem Zeitpunk';
#

#
SELECT CURRENT_TIME();
SELECT 'tmp_txp_pre:';
SELECT 'Entfernen einer eventuell schon existierenden Tabelle tmp_txp_pre';
#
DROP TABLE
  IF EXISTS
  tmp_txp_pre
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_txp_pre:';
SELECT 'Erstellen der Tabelle tmp_txp_pre';
#
CREATE TABLE
  IF NOT EXISTS
  tmp_txp_pre (
    id                 INT UNSIGNED  NOT NULL AUTO_INCREMENT  KEY
  , time_id            INT
  , page_id            INT
  , namespace_id       INT
  , rev_count_pre_id   INT
  , rev_count_pre      INT
  , rev_last_time_stamp BIGINT
  , rev_last_id        INT
  , rev_last_len_id    INT
  , rev_last_len       INT
  )
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_txp_pre:';
SELECT 'Anzahl der Revisionen vor einem Zeitpunkt (rev_count_pre)';
SELECT 'und Zeitpunkt der letzten Revision vor einem Zeitpunkt (rev_last_time_stamp)';
SELECT 'ueber die Seiten und die Zeit ermitteln.';
#
INSERT INTO
  tmp_txp_pre (
    time_id
  , page_id
  , namespace_id
  , rev_count_pre
  , rev_last_time_stamp
  )
SELECT
  tmp_time.id
, tmp_page.id
, tmp_page.namespace_id
, COUNT(tmp_rev.id)
, MAX(tmp_rev.time_stamp)
FROM (
  tmp_time
  JOIN (
    tmp_page
    JOIN
      tmp_rev
    ON
      tmp_rev.page_id = tmp_page.id
  )
  ON
    tmp_rev.time_id < tmp_time.id
)
```

```
GROUP BY
  tmp_time.id
, tmp_page.id
;
CREATE UNIQUE INDEX txp_id ON tmp_txp_pre (time_id, page_id);
CREATE INDEX time_id ON tmp_txp_pre (time_id);
CREATE INDEX page_id ON tmp_txp_pre (page_id);
CREATE INDEX namespace_id ON tmp_txp_pre (namespace_id);
CREATE INDEX rev_count_in ON tmp_txp_pre (rev_count_pre);
CREATE INDEX rev_last_time_stamp ON tmp_txp_pre (rev_last_time_stamp);


#
SELECT CURRENT_TIME();
SELECT 'tmp_txp_pre:';
SELECT 'Nummer der letzten Revision vor einem Zeitraum (rev_last_id)';
SELECT 'ueber die Seiten und die Zeit ermitteln.';
#

UPDATE
  tmp_txp_pre
SET
  tmp_txp_pre.rev_last_id = (
    SELECT
      MAX(tmp_rev.id)
    FROM
      tmp_rev
    WHERE
      tmp_rev.time_stamp = tmp_txp_pre.rev_last_time_stamp
  )
;
CREATE INDEX rev_last_id ON tmp_txp_pre (rev_last_id);


#
SELECT CURRENT_TIME();
SELECT 'tmp_txp_pre:';
SELECT 'Klassifizierung der Anzahl der Revisionen (rev_count_pre_id)';
#

UPDATE
  tmp_txp_pre
, tmp_class
SET
  tmp_txp_pre.rev_count_pre_id = tmp_class.id
WHERE
  tmp_txp_pre.rev_count_pre BETWEEN tmp_class.lim_min AND tmp_class.lim_max
;
CREATE INDEX  rev_count_pre_id ON  tmp_txp_pre (rev_count_pre_id);


#
SELECT CURRENT_TIME();
SELECT 'tmp_txp_pre:';
SELECT 'Texte-Laenge und Klassifizierung der Text-Laenge';
SELECT 'der letzten Revision vor einem Zeitraum';
SELECT 'ueber die Seiten und die Zeit aus tmp_rev kopieren.';
#

UPDATE
  tmp_txp_pre
, tmp_rev
SET
  tmp_txp_pre.rev_last_len = tmp_rev.text_len
, tmp_txp_pre.rev_last_len_id = tmp_rev.text_len_id
WHERE
  tmp_rev.id = tmp_txp_pre.rev_last_id
;
CREATE INDEX  rev_last_len ON  tmp_txp_pre (rev_last_len);
CREATE INDEX  rev_last_len_id ON  tmp_txp_pre (rev_last_len_id);




#
SELECT CURRENT_TIME();
SELECT 'tmp_txu_in:';
SELECT 'Kreuztabelle der Benutzer ueber die Zeit (tmp_txu_in)';
SELECT 'zur Auswertung innerhalb der Zeitintervalle';
#

#
SELECT CURRENT_TIME();
SELECT 'tmp_txu_in:';
```

```
SELECT 'Entfernen einer eventuell schon existierenden Tabelle tmp_txu_in';
#
DROP TABLE
  IF EXISTS
  tmp_txu_in
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_txu_in:';
SELECT 'Erstellen der Tabelle tmp_txu_in';
#
CREATE TABLE
  IF NOT EXISTS
  tmp_txu_in (
    id                INT UNSIGNED  NOT NULL AUTO_INCREMENT  KEY
  , time_id           INT
  , user_id           INT
  , rev_count_in_id   INT
  , rev_count_in      INT
  )
;


#
SELECT CURRENT_TIME();
SELECT 'tmp_txu_in:';
SELECT 'Anzahl der Revisionen innerhalb eines Zeitintervalls (rev_count_in)';
SELECT 'ueber die Benutzer und die Zeit ermitteln.';
#

INSERT INTO
  tmp_txu_in (
    time_id
  , user_id
  , rev_count_in
  )
SELECT
  tmp_time.id
, tmp_user.id
, COUNT(tmp_rev.id)
FROM (
  tmp_time
  JOIN (
    tmp_user
    JOIN
      tmp_rev
    ON
      tmp_rev.user_id = tmp_user.id
  )
  ON
    tmp_rev.time_id = tmp_time.id
)
GROUP BY
  tmp_time.id
, tmp_user.id
;
CREATE UNIQUE INDEX txu_id ON tmp_txu_in (time_id, user_id);
CREATE INDEX time_id ON tmp_txu_in (time_id);
CREATE INDEX user_id ON tmp_txu_in (user_id);
CREATE INDEX rev_count_in ON tmp_txu_in (rev_count_in);

#
SELECT CURRENT_TIME();
SELECT 'tmp_txu_in:';
SELECT 'Klassifizierung der Anzahl der Revisionen (rev_count_in_id)';
#

UPDATE
  tmp_txu_in
, tmp_class
SET
  tmp_txu_in.rev_count_in_id = tmp_class.id
WHERE
  tmp_txu_in.rev_count_in BETWEEN tmp_class.lim_min AND tmp_class.lim_max
;
CREATE INDEX  rev_count_in_id ON  tmp_txu_in (rev_count_in_id);


#
SELECT CURRENT_TIME();
SELECT 'tmp_txu_pre:';
SELECT 'Kreuztabelle der Benutzer ueber die Zeit (tmp_txu_pre)';
```

```
SELECT 'zur Auswertung vor einem Zeitpunk';
#

#
SELECT CURRENT_TIME();
SELECT 'tmp_txu_pre:';
SELECT 'Entfernen einer eventuell schon existierenden Tabelle tmp_txu_pre';
#
DROP TABLE
  IF EXISTS
  tmp_txu_pre
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_txu_pre:';
SELECT 'Erstellen der Tabelle tmp_txu_pre';
#
CREATE TABLE
  IF NOT EXISTS
  tmp_txu_pre (
    id                INT UNSIGNED  NOT NULL AUTO_INCREMENT  KEY
  , time_id           INT
  , user_id           INT
  , rev_count_pre_id  INT
  , rev_count_pre     INT
  )
;


#
SELECT CURRENT_TIME();
SELECT 'tmp_txu_pre:';
SELECT 'Anzahl der Revisionen vor einem Zeitpunkt (rev_count_pre)';
SELECT 'ueber die Benutzer und die Zeit ermitteln.';
#

INSERT INTO
  tmp_txu_pre (
    time_id
  , user_id
  , rev_count_pre
  )
SELECT
  tmp_time.id
, tmp_user.id
, COUNT(tmp_rev.id)
FROM (
  tmp_time
  JOIN (
    tmp_user
    JOIN
      tmp_rev
    ON
      tmp_rev.user_id = tmp_user.id
  )
  ON
    tmp_rev.time_id < tmp_time.id
)
GROUP BY
  tmp_time.id
, tmp_user.id
;
CREATE UNIQUE INDEX txu_id ON tmp_txu_pre (time_id, user_id);
CREATE INDEX time_id ON tmp_txu_pre (time_id);
CREATE INDEX user_id ON tmp_txu_pre (user_id);
CREATE INDEX rev_count_in ON tmp_txu_pre (rev_count_pre);


#
SELECT CURRENT_TIME();
SELECT 'tmp_txu_pre:';
SELECT 'Klassifizierung der Anzahl der Revisionen (rev_count_pre_id)';
#

UPDATE
  tmp_txu_pre
, tmp_class
SET
  tmp_txu_pre.rev_count_pre_id = tmp_class.id
WHERE
  tmp_txu_pre.rev_count_pre BETWEEN tmp_class.lim_min AND tmp_class.lim_max
;
CREATE INDEX  rev_count_pre_id ON  tmp_txu_pre (rev_count_pre_id);
```

```
#
SELECT CURRENT_TIME();
SELECT 'Ende der Abfrage!!!'
#
```

Auswertung_Step_04.sql

```
# MySQL 5.0.24

#
SELECT CURRENT_TIME();
SELECT 'Quantitative Auswertung der Wikipedia-Datenbank';
SELECT 'Abfrage Auswertung_Step_04.sql';
#
# Diese Abfrage ist die dritte von sechs Abfragen, die die statistische Auswertung
# eines als mysql-Datenbank vorliegenden Wikipedia-Dumps ermoeglichen.
# Fuer die korreke Funkionsweise sollten die Abfragen in folgender Reihenfolge
# ausgefuehrt werden:
#
# 1. Auswertung_Step_01.sql # tmp_time, tmp_class
# 2. Auswertung_Step_02.sql # tmp_page, tmp_rev, tmp_user
# 3. Auswertung_Step_03.sql # tmp_txp_in, tmp_txp_pre, tmp_txu_in, tmp_txu_pre
# 4. Auswertung_Step_04.sql # Ausgabe der Informationen ueber die Zeit
# 5. Auswertung_Step_05.sql # Ausgabe der gesamt Informationen
# 6. Auswertung_Step_06.sql # Ausgabe der Listen
#
#
# In dieser Abfrage erstellt eine Ausgabe der Informationen
# aus den in Auswertung_Step_03.sql erstellten Kreuztabellen:
#
# - tmp_txp_in: Informationen ueber die Seiten innerhalb eines Zeitintervalls
# - tmp_txp_pre: Informationen ueber die Seiten vor einem Zeitpunkt
# - tmp_txu_in: Informationen ueber die Benutzer innerhalb eines Zeitintervalls
# - tmp_txu_pre: Informationen ueber die Benutzer vor einem Zeitpunkt
#
#
# Weitere Erlaeuterungen zum Einsatz dieser mysql-Abfrage:
#   "Quantitative Auswertung der Wikipedia-Datenbank"
#   von Christoph Hassel & Andreas Seier, 2006
#   http://aseier.de/wikipedia/
#


#
SELECT CURRENT_TIME();
SELECT 'tmp_txp_in:';
SELECT 'Kreuztabelle der Seiten ueber die Zeit (tmp_txp_in)';
SELECT 'zur Auswertung innerhalb der Zeitintervalle';
#

#
SELECT 'tmp_txp_in:';
SELECT 'Anzahl Revisionen (SUM(rev_count_in)),';
SELECT 'Anzahl Seiten (count(page_id))';
SELECT 'und Durchschnittliche Anzahl Revisionen (AVG(rev_count_in) der Seiten,';
SELECT 'die innerhalb eines Zeitintervalls (time_id) bearbeitet wurden,';
SELECT 'ueber die Zeit (time_id) ausgeben';
SELECT 'AVG(rev_count_in) = SUM(rev_count_in) / count(page_id)';
#
SELECT time_id, SUM(rev_count_in), COUNT(page_id), AVG(rev_count_in)
  FROM tmp_txp_in
  GROUP BY time_id
;

#
SELECT 'tmp_txp_in:';
SELECT 'Anzahl Seiten (count(page_id))';
SELECT 'ueber die Klassen der Anzahl der Revisionen';
SELECT 'innerhalb eines Zeitraums (rev_count_in_id)';
SELECT 'und die Zeit (time_id) ausgeben';
#
SELECT
  time_id
, rev_count_in_id
, count(page_id)
FROM
  tmp_txp_in
GROUP BY
```

```
  time_id
, rev_count_in_id
;


#
SELECT CURRENT_TIME();
SELECT 'tmp_txp_pre:';
SELECT 'Kreuztabelle der Seiten ueber die Zeit (tmp_txp_pre)';
SELECT 'zur Auswertung vor einem Zeitpunkt';
#

#
SELECT 'tmp_txp_pre:';
SELECT 'Anzahl Revisionen (SUM(rev_count_pre)),';
SELECT 'Anzahl Seiten (count(page_id)) und';
SELECT 'Durchschnittliche Anzahl Revisionen (AVG(rev_count_pre) der Seiten,';
SELECT 'die vor einem Zeitpunkt (time_id) bearbeitet wurden,';
SELECT 'ueber die Zeit (time_id) ausgeben';
SELECT 'AVG(rev_count_pre) = SUM(rev_count_pre) / count(page_id)';
#
SELECT time_id, SUM(rev_count_pre), COUNT(page_id), AVG(rev_count_pre)
  FROM tmp_txp_pre
  GROUP BY time_id
;

#
SELECT 'tmp_txp_pre:';
SELECT 'Anzahl Seiten (count(page_id))';
SELECT 'ueber die Klassen der Anzahl der Revisionen';
SELECT 'vor einem Zeitpunkt (rev_count_pre_id)';
SELECT 'und die Zeit (time_id) ausgeben';
#
SELECT
  time_id
, rev_count_pre_id
, count(page_id)
FROM
  tmp_txp_pre
GROUP BY
  time_id
, rev_count_pre_id
;

#
SELECT 'tmp_txp_pre:';
SELECT 'Gesamt-Text-Laenge (SUM(rev_last_len)) der Seiten,';
SELECT 'Anzahl Seiten (count(page_id)) und';
SELECT 'Durchschnittliche Text-Laenge (AVG(rev_last_len) der Seiten,';
SELECT 'die vor einem Zeitpunkt (time_id) bearbeitet wurden,';
SELECT 'ueber die Zeit (time_id) ausgeben';
SELECT 'AVG(rev_last_len) = SUM(rev_last_len) / count(page_id)';
#
SELECT time_id, SUM(rev_last_len), count(page_id), AVG(rev_last_len)
  FROM tmp_txp_pre
  GROUP BY time_id
;

#
SELECT 'tmp_txp_pre:';
SELECT 'Anzahl Seiten (count(page_id))';
SELECT 'ueber die Klassen der Text-Laenge der letzten Revision ';
SELECT 'vor einem Zeitpunkt (rev_count_pre_id)';
SELECT 'und die Zeit (time_id) ausgeben';
#
SELECT
  time_id
, rev_last_len_id
, count(page_id)
FROM
  tmp_txp_pre
GROUP BY
  time_id
, rev_last_len_id
;


#
SELECT CURRENT_TIME();
SELECT 'tmp_txu_in:';
SELECT 'Kreuztabelle der Benutzer ueber die Zeit (tmp_txu_in)';
SELECT 'zur Auswertung innerhalb der Zeitintervalle';
#
```

```
#
SELECT 'tmp_txu_in:';
SELECT 'Anzahl Revisionen (SUM(rev_count_in)),';
SELECT 'Anzahl Benutzer (count(user_id)) und';
SELECT 'Durchschnittliche Anzahl Revisionen (AVG(rev_count_in) der Benutzer,';
SELECT 'die innerhalb eines Zeitintervalls (time_id) aktiv waren,';
SELECT 'ueber die Zeit (time_id) ausgeben';
SELECT 'AVG(rev_count_in) = SUM(rev_count_in) / count(user_id)';
#
SELECT time_id, SUM(rev_count_in), COUNT(user_id), AVG(rev_count_in)
  FROM tmp_txu_in
  GROUP BY time_id
;

#
SELECT 'tmp_txu_in:';
SELECT 'Anzahl Benutzer (count(user_id))';
SELECT 'ueber die Klassen der Anzahl der Revisionen';
SELECT 'innerhalb eines Zeitraums (rev_count_in_id)';
SELECT 'und die Zeit (time_id) ausgeben';
#
SELECT
  time_id
, rev_count_in_id
, count(user_id)
FROM
  tmp_txu_in
GROUP BY
  time_id
, rev_count_in_id
;


#
SELECT CURRENT_TIME();
SELECT 'tmp_txu_pre:';
SELECT 'Kreuztabelle der Benutzer ueber die Zeit (tmp_txu_pre)';
SELECT 'zur Auswertung vor einem Zeitpunkt';
#

#
SELECT 'tmp_txu_pre:';
SELECT 'Anzahl Revisionen (SUM(rev_count_pre)),';
SELECT 'Anzahl Benutzer (count(user_id)) und';
SELECT 'Durchschnittliche Anzahl Revisionen (AVG(rev_count_pre) der Benutzer,';
SELECT 'die innerhalb eines Zeitintervalls (time_id) aktiv waren,';
SELECT 'ueber die Zeit (time_id) ausgeben';
SELECT 'AVG(rev_count_pre) = SUM(rev_count_pre) / count(user_id)';
#
SELECT time_id, SUM(rev_count_pre), COUNT(user_id), AVG(rev_count_pre)
  FROM tmp_txu_pre
  GROUP BY time_id
;

#
SELECT 'tmp_txu_pre:';
SELECT 'Anzahl Benutzer (count(user_id))';
SELECT 'ueber die Klassen der Anzahl der Revisionen';
SELECT 'vor einem Zeitpunkt (rev_count_pre_id)';
SELECT 'und die Zeit (time_id) ausgeben';
#
SELECT
  time_id
, rev_count_pre_id
, count(user_id)
FROM
  tmp_txu_pre
GROUP BY
  time_id
, rev_count_pre_id
;


#
SELECT CURRENT_TIME();
SELECT 'Ende der Abfrage!!!'
#
```

Auswertung_Step_05.sql

```
# MySQL 5.0.24
```

```
#
SELECT CURRENT_TIME();
SELECT 'Quantitative Auswertung der Wikipedia-Datenbank';
SELECT 'Abfrage Auswertung_Step_05.sql';
#
# Diese Abfrage ist die dritte von sechs Abfragen, die die statistische Auswertung
# eines als mysql-Datenbank vorliegenden Wikipedia-Dumps ermoeglichen.
# Fuer die korreke Funkionsweise sollten die Abfragen in folgender Reihenfolge
# ausgefuehrt werden:
#
# 1. Auswertung_Step_01.sql # tmp_time, tmp_class
# 2. Auswertung_Step_02.sql # tmp_page, tmp_rev, tmp_user
# 3. Auswertung_Step_03.sql # tmp_txp_in, tmp_txp_pre, tmp_txu_in, tmp_txu_pre
# 4. Auswertung_Step_04.sql # Ausgabe der Informationen ueber die Zeit
# 5. Auswertung_Step_05.sql # Ausgabe der gesamt Informationen
# 6. Auswertung_Step_06.sql # Ausgabe der Listen
#
#
# In dieser Abfrage erstellt eine Ausgabe der Informationen
# aus den in Auswertung_Step_02.sql erstellten Tabellen:
#
# - tmp_page
# - tmp_rev
# - tmp_user
#
#
# Weitere Erlaeuterungen zum Einsatz dieser mysql-Abfrage:
#   "Quantitative Auswertung der Wikipedia-Datenbank"
#   von Christoph Hassel & Andreas Seier, 2006
#   http://aseier.de/wikipedia/
#


#
SELECT CURRENT_TIME();
SELECT 'tmp_page:'
SELECT 'Erste Seite';
#
SELECT wp2.page.page_id, wp2.page.page_title
  FROM wp2.page, tmp_rev
  WHERE
    wp2.page.page_id = tmp_rev.page_id
    AND tmp_rev.time_stamp =
      (SELECT MIN(tmp_rev.time_stamp) FROM tmp_rev)
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_page:'
SELECT 'Anzahl neuer Seiten im Zeitintervall';
#
SELECT rev_first_time_id, COUNT(id)
  FROM tmp_page
  GROUP BY rev_first_time_id
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_rev:'
SELECT 'Anzahl Revisionen im Zeitintervall';
#
SELECT time_id, COUNT(id)
  FROM tmp_rev
  GROUP BY time_id
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_rev:'
SELECT 'Anzahl Revisionen durch registrierte Benutzer im Zeitintervall';
#
SELECT time_id, COUNT(id)
  FROM tmp_rev
  WHERE NOT user_anonym
  GROUP BY time_id
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_rev:'
SELECT 'Anzahl Revisionen durch anonyme Benutzer im Zeitintervall';
```

```
#
SELECT time_id, COUNT(id)
  FROM tmp_rev
  WHERE user_anonym
  GROUP BY time_id
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_user:'
SELECT 'Anzahl neuer Benutzer im Zeitintervall';
#
SELECT rev_first_time_id, COUNT(id)
  FROM tmp_user
  GROUP BY rev_first_time_id
;

#
SELECT CURRENT_TIME();
SELECT 'tmp_user:'
SELECT 'Anzahl Benutzer die vor und nach oder im Zeitintervall aktiv waren';
#
SELECT tmp_time.id, COUNT(tmp_user.id)
  FROM tmp_time JOIN tmp_user
  ON tmp_time.id BETWEEN tmp_user.rev_first_time_id AND tmp_user.rev_last_time_id
  GROUP BY tmp_time.id
;


#
SELECT CURRENT_TIME();
SELECT 'Ende der Abfrage!!!'
#
```

Auswertung_Step_06.sql

```
# MySQL 5.0.24

#
SELECT CURRENT_TIME();
SELECT 'Quantitative Auswertung der Wikipedia-Datenbank';
SELECT 'Abfrage Auswertung_Step_06.sql';
#
# Diese Abfrage ist die dritte von sechs Abfragen, die die statistische Auswertung
# eines als mysql-Datenbank vorliegenden Wikipedia-Dumps ermoeglichen.
# Fuer die korreke Funkionsweise sollten die Abfragen in folgender Reihenfolge
# ausgefuehrt werden:
#
# 1. Auswertung_Step_01.sql # tmp_time, tmp_class
# 2. Auswertung_Step_02.sql # tmp_page, tmp_rev, tmp_user
# 3. Auswertung_Step_03.sql # tmp_txp_in, tmp_txp_pre, tmp_txu_in, tmp_txu_pre
# 4. Auswertung_Step_04.sql # Ausgabe der Informationen ueber die Zeit
# 5. Auswertung_Step_05.sql # Ausgabe der gesamt Informationen
# 6. Auswertung_Step_06.sql # Ausgabe der Listen
#
#
# In dieser Abfrage erstellt eine Ausgabe der Informationen
# aus den in Auswertung_Step_03.sql erstellten Kreuztabellen:
#
# - tmp_txp_in: Informationen ueber die Seiten innerhalb eines Zeitintervalls
# - tmp_txp_pre: Informationen ueber die Seiten vor einem Zeitpunkt
# - tmp_txu_in: Informationen ueber die Benutzer innerhalb eines Zeitintervalls
# - tmp_txu_pre: Informationen ueber die Benutzer vor einem Zeitpunkt
#
#
# Weitere Erlaeuterungen zum Einsatz dieser mysql-Abfrage:
#   "Quantitative Auswertung der Wikipedia-Datenbank"
#   von Christoph Hassel & Andreas Seier, 2006
#   http://aseier.de/wikipedia/
#


#
SELECT CURRENT_TIME();
SELECT 'tmp_txp_pre:'
SELECT 'Absteigende Liste der zehn Seiten mit den meisten Revisionen vor einem Zeitpunkt';
#
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 1
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
```

```
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 2
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 3
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 4
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 5
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 6
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 7
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 8
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 9
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 10
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 11
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 12
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 13
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 14
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 15
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 16
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 17
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 18
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 19
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 20
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 21
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 22
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 23
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 24
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 25
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 26
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 27
ORDER BY rev_count_pre DESC LIMIT 10 ;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_count_pre FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 28
```

```
ORDER BY rev_count_pre DESC LIMIT 10 ;

#
SELECT CURRENT_TIME();
SELECT 'tmp_txp_pre:'
SELECT 'Absteigende Liste der zehn Seiten mit den laengsten Texten zu einem Zeitpunkt';
#
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 1
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 2
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 3
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 4
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 5
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 6
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 7
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 8
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 9
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 10
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 11
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 12
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 13
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 14
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 15
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 16
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 17
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 18
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 19
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 20
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 21
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 22
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 23
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 24
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
```

```
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 25
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 26
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 27
ORDER BY rev_last_len DESC LIMIT 10;
SELECT tmp_txp_pre.time_id, wp2.page.page_title, tmp_txp_pre.rev_last_len FROM tmp_txp_pre, wp2.page
WHERE tmp_txp_pre.page_id = wp2.page.page_id AND tmp_txp_pre.time_id = 28
ORDER BY rev_last_len DESC LIMIT 10;

#
SELECT CURRENT_TIME();
SELECT 'tmp_txp_in:'
SELECT 'Absteigende Liste der zehn Seiten mit den meisten Revisionen innerhalb eines Zeitintervalls';
#
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 1
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 2
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 3
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 4
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 5
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 6
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 7
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 8
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 9
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 10
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 11
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 12
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 13
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 14
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 15
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 16
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 17
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 18
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 19
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 20
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 21
ORDER BY rev_count_in DESC LIMIT 10;
```

```
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 22
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 23
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 24
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 25
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 26
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 27
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txp_in.time_id, wp2.page.page_title, tmp_txp_in.rev_count_in FROM tmp_txp_in, wp2.page
WHERE tmp_txp_in.page_id = wp2.page.page_id AND tmp_txp_in.time_id = 28
ORDER BY rev_count_in DESC LIMIT 10;

#
SELECT CURRENT_TIME();
SELECT 'tmp_txu_pre:'
SELECT 'Absteigende Liste der Benutzer mit den meisten Revisionen vor einem Zeitpunkt';
#
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 1
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 2
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 3
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 4
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 5
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 6
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 7
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 8
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 9
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 10
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 11
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 12
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 13
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 14
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 15
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 16
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 17
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 18
```

```
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 19
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 20
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 21
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 22
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 23
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 24
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 25
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 26
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 27
ORDER BY rev_count_pre DESC LIMIT 10;
SELECT tmp_txu_pre.time_id, tmp_user.name, tmp_txu_pre.rev_count_pre FROM tmp_txu_pre, tmp_user
WHERE tmp_txu_pre.user_id = tmp_user.id AND tmp_txu_pre.time_id = 28
ORDER BY rev_count_pre DESC LIMIT 10;

#
SELECT CURRENT_TIME();
SELECT 'tmp_txu_in:'
SELECT 'Absteigende Liste der Benutzer mit den meisten Revisionen innerhalb eines Zeitintervalls';
#
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 1
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 2
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 3
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 4
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 5
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 6
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 7
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 8
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 9
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 10
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 11
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 12
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 13
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 14
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
```

```
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 15
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 16
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 17
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 18
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 19
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 20
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 21
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 22
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 23
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 24
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 25
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 26
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 27
ORDER BY rev_count_in DESC LIMIT 10;
SELECT tmp_txu_in.time_id, tmp_user.name, tmp_txu_in.rev_count_in FROM tmp_txu_in, tmp_user
WHERE tmp_txu_in.user_id = tmp_user.id AND tmp_txu_in.time_id = 28
ORDER BY rev_count_in DESC LIMIT 10;


#
SELECT CURRENT_TIME();
SELECT 'Ende der Abfrage!!!'
#
```

Materialien

[DE Wikipedia Database Dump] Sicherung der Wikipedia Datenbank in der Version vom 04.06.2006, http://download.wikimedia.org/dewiki/20060604/dewiki-20060604-pages-meta-history.xml.7z (Recherche: 08.08.2006; Datei ist nicht mehr verfügbar).

[Doering 2003] Nicola Doering: *Sozialpsychologie des Internet. Die Bedeutung des Internet für Kommunikationsprozesse, Identitäten, soziale Beziehungen und Gruppen*, 2., vollständig überarbeitete und erweiterte Auflage, Göttingen u.a.

[MySQL Manual] MySQL AB (Hrsg.): *MySQL 5.0 Reference Manual*, URL: http://dev.mysql.com/doc/refman/5.0/en/, Recherche: 08.02.2007.

[Stegbauer 2006] Christian Stegbauer: *Von den Online Communities zu den computervermittelten sozialen Netzwerken. Eine Reinterpretation klassischer Studien*, in: Christian Stegbauer und Alexander Rausch (Hrsg.): Strukturalistische Internetforschung. Netzwerkanalysen internetbasierter Kommunikationsräume, Wiesbaden, S. 67–94.

[Voß 2005] Jakob Voß: *Measuring Wikipedia*, in: Proceedings of the 10th International Conference of the International Society for Scientometrics and Informetrics, URL: eprints.rclis.org/archive/00003610/01/MeasuringWikipedia2005.pdf, Recherche: 3.2.2007.

[Wikimetrics] *Wikimetrics. a wiki research blog* http://wm.sieheauch.de/, Recherche: 22.02.2007.

[WM-Meta-DD] Wikimedia Foundation: *Data Dumps*, URL: http://meta.wikimedia.org/w/index.php?title=Data_dumps&oldid=498820, Recherche: 10.01.2007.

[Zachte 2007] Erik Zachte: *Wikistats*, URL: http://meta.wikimedia.org/w/index.php?title=Wikistats&oldid=517340, Recherche: 22.02.2007.

Die Skripte stehen unter der URL http://aseier.de/wikipedia/Skripte/ zum Download zur Verfügung.

Lizenz