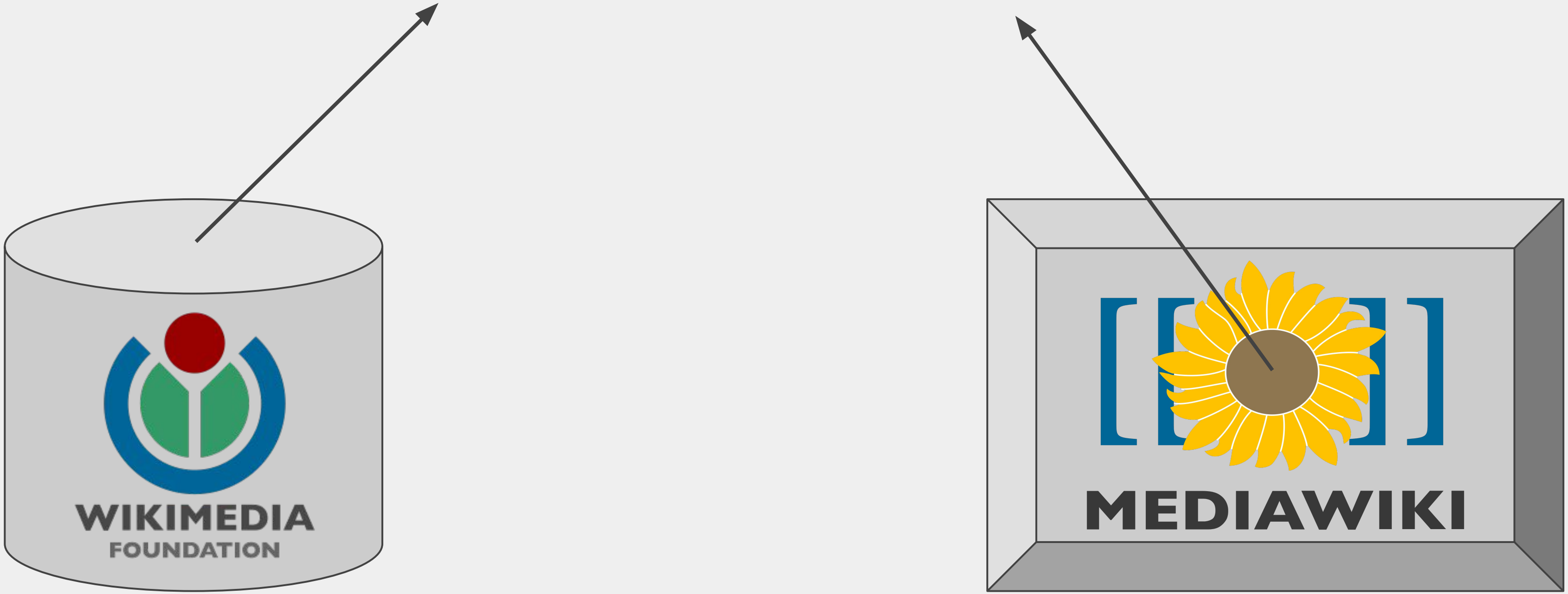
The English Wikipedia Database

Paolo Tamagnini
Benedetta Checcarelli

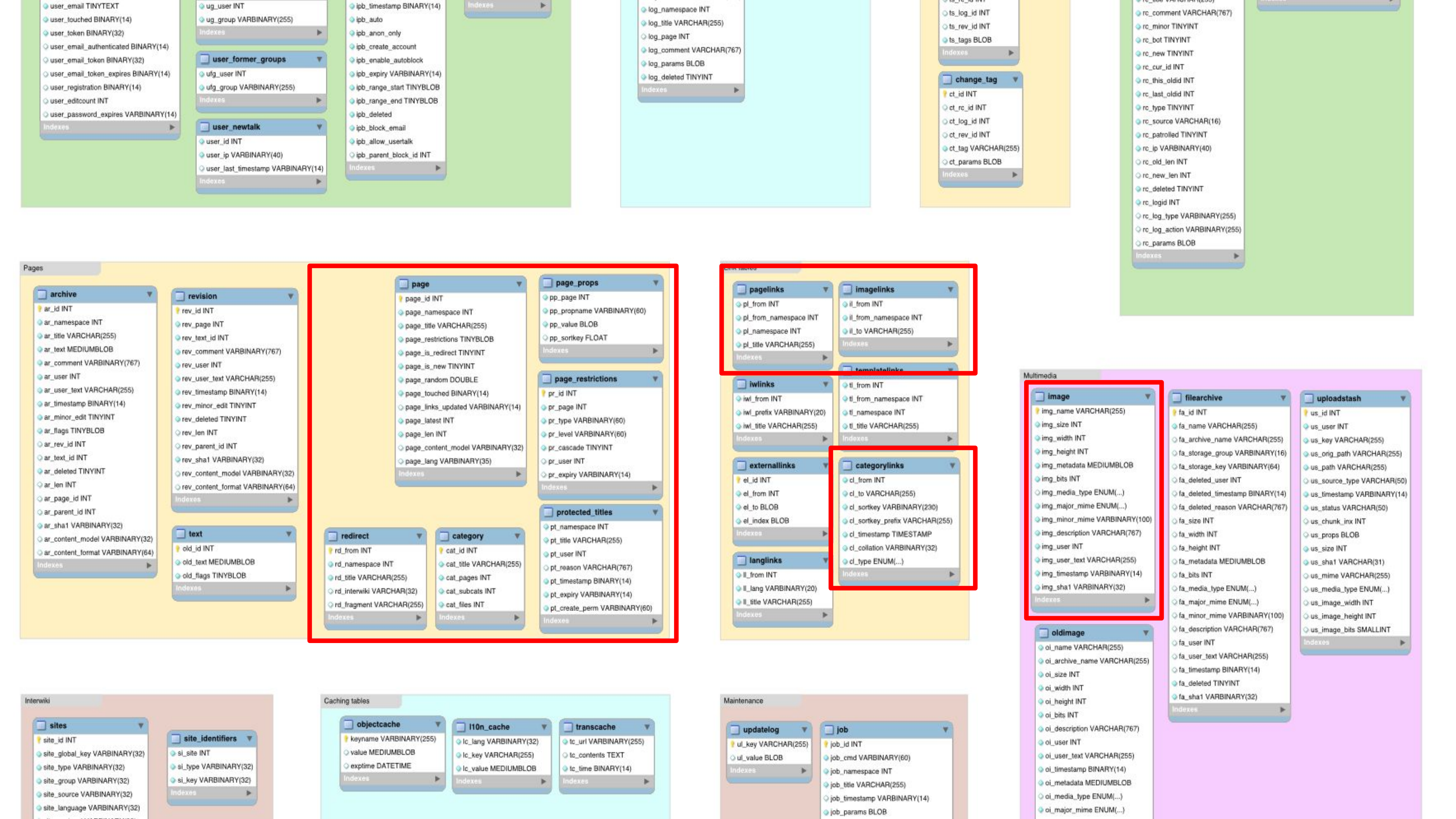
WIKIPEDIA
The Free Encyclopedia

Local database of some tables of the whole schema, containing most of the data of the today Wikipedia



Data content from english Wikipedia

Database MySQL infrastructure



user_email TINYTEXT
user_touched BINARY(14)
user_token BINARY(32)
user_email_authenticated BINARY(14)
user_email_token BINARY(32)
user_email_token_expires BINARY(14)
user_registration BINARY(14)
user_editcount INT
user_password_expires VARBINARY(14)
Indexes

ug_user INT
ug_group VARBINARY(255)
Indexes

user_former_groups
ufg_user INT
ufg_group VARBINARY(255)
Indexes

user_newtalk
user_id INT
user_ip VARBINARY(40)
user_last_timestamp VARBINARY(14)
Indexes

ipb_timestamp BINARY(14)
ipb_auto
ipb_anon_only
ipb_create_account
ipb_enable_autoblock
ipb_expiry VARBINARY(14)
ipb_range_start TINYBLOB
ipb_range_end TINYBLOB
ipb_deleted
ipb_block_email
ipb_allow_usertalk
ipb_parent_block_id INT
Indexes

Indexes

log_namespace INT
log_title VARCHAR(255)
log_page INT
log_comment VARCHAR(767)
log_params BLOB
log_deleted TINYINT
Indexes

ts_log_id INT
ts_rev_id INT
ts_tags BLOB
Indexes

change_tag
ct_id INT
ct_rc_id INT
ct_log_id INT
ct_rev_id INT
ct_tag VARCHAR(255)
ct_params BLOB
Indexes

rc_comment VARCHAR(767)
rc_minor TINYINT
rc_bot TINYINT
rc_new TINYINT
rc_cur_id INT
rc_this_oldid INT
rc_last_oldid INT
rc_type TINYINT
rc_source VARCHAR(16)
rc_patrolled TINYINT
rc_ip VARBINARY(40)
rc_old_len INT
rc_new_len INT
rc_deleted TINYINT
rc_logid INT
rc_log_type VARBINARY(255)
rc_log_action VARBINARY(255)
rc_params BLOB
Indexes

Pages

archive
ar_id INT
ar_namespace INT
ar_title VARCHAR(255)
ar_text MEDIUMBLOB
ar_comment VARBINARY(767)
ar_user INT
ar_user_text VARCHAR(255)
ar_timestamp BINARY(14)
ar_minor_edit TINYINT
ar_flags TINYBLOB
ar_rev_id INT
ar_text_id INT
ar_deleted TINYINT
ar_len INT
ar_page_id INT
ar_parent_id INT
ar_sha1 VARBINARY(32)
ar_content_model VARBINARY(32)
ar_content_format VARBINARY(64)
Indexes

revision
rev_id INT
rev_page INT
rev_text_id INT
rev_comment VARBINARY(767)
rev_user INT
rev_user_text VARCHAR(255)
rev_timestamp BINARY(14)
rev_minor_edit TINYINT
rev_deleted TINYINT
rev_len INT
rev_parent_id INT
rev_sha1 VARBINARY(32)
rev_content_model VARBINARY(32)
rev_content_format VARBINARY(64)
Indexes

text
old_id INT
old_text MEDIUMBLOB
old_flags TINYBLOB
Indexes

page
page_id INT
page_namespace INT
page_title VARCHAR(255)
page_restrictions TINYBLOB
page_is_redirect TINYINT
page_is_new TINYINT
page_random DOUBLE
page_touched BINARY(14)
page_links_updated VARBINARY(14)
page_latest INT
page_len INT
page_content_model VARBINARY(32)
page_lang VARBINARY(35)
Indexes

page_props
pp_page INT
pp_propname VARBINARY(60)
pp_value BLOB
pp_sortkey FLOAT
Indexes

page_restrictions
pr_id INT
pr_page INT
pr_type VARBINARY(60)
pr_level VARBINARY(60)
pr_cascade TINYINT
pr_user INT
pr_expiry VARBINARY(14)
Indexes

redirect
rd_from INT
rd_namespace INT
rd_title VARCHAR(255)
rd_interwiki VARCHAR(32)
rd_fragment VARCHAR(255)
Indexes

category
cat_id INT
cat_title VARCHAR(255)
cat_pages INT
cat_subcats INT
cat_files INT
Indexes

protected_titles
pt_namespace INT
pt_title VARCHAR(255)
pt_user INT
pt_reason VARCHAR(767)
pt_timestamp BINARY(14)
pt_expiry VARBINARY(14)
pt_create_perm VARBINARY(60)
Indexes

Link tables

pagelinks
pl_from INT
pl_from_namespace INT
pl_namespace INT
pl_title VARCHAR(255)
Indexes

imagelinks
il_from INT
il_from_namespace INT
il_to VARCHAR(255)
Indexes

iwlinks
iwl_from INT
iwl_prefix VARBINARY(20)
iwl_title VARCHAR(255)
Indexes

templatelinks
tl_from INT
tl_from_namespace INT
tl_namespace INT
tl_title VARCHAR(255)
Indexes

externallinks
el_id INT
el_from INT
el_to BLOB
el_index BLOB
Indexes

categorylinks
cl_from INT
cl_to VARCHAR(255)
cl_sortkey VARBINARY(230)
cl_sortkey_prefix VARCHAR(255)
cl_timestamp TIMESTAMP
cl_collation VARBINARY(32)
cl_type ENUM(...)
Indexes

langlinks
ll_from INT
ll_lang VARBINARY(20)
ll_title VARCHAR(255)
Indexes

Multimedia

image
img_name VARCHAR(255)
img_size INT
img_width INT
img_height INT
img_metadata MEDIUMBLOB
img_bits INT
img_media_type ENUM(...)
img_major_mime ENUM(...)
img_minor_mime VARBINARY(100)
img_description VARCHAR(767)
img_user INT
img_user_text VARCHAR(255)
img_timestamp VARBINARY(14)
img_sha1 VARBINARY(32)
Indexes

oldimage
oi_name VARCHAR(255)
oi_archive_name VARCHAR(255)
oi_size INT
oi_width INT
oi_height INT
oi_bits INT
oi_description VARCHAR(767)
oi_user INT
oi_user_text VARCHAR(255)
oi_timestamp BINARY(14)
oi_metadata MEDIUMBLOB
oi_media_type ENUM(...)
oi_major_mime ENUM(...)

filearchive
fa_id INT
fa_name VARCHAR(255)
fa_archive_name VARCHAR(255)
fa_storage_group VARBINARY(16)
fa_storage_key VARBINARY(64)
fa_deleted_user INT
fa_deleted_timestamp BINARY(14)
fa_deleted_reason VARCHAR(767)
fa_size INT
fa_width INT
fa_height INT
fa_metadata MEDIUMBLOB
fa_bits INT
fa_media_type ENUM(...)
fa_major_mime ENUM(...)
fa_minor_mime VARBINARY(100)
fa_description VARCHAR(767)
fa_user INT
fa_user_text VARCHAR(255)
fa_timestamp BINARY(14)
fa_deleted TINYINT
fa_sha1 VARBINARY(32)
Indexes

uploadstash
us_id INT
us_user INT
us_key VARCHAR(255)
us_orig_path VARCHAR(255)
us_path VARCHAR(255)
us_source_type VARCHAR(50)
us_timestamp VARBINARY(14)
us_status VARCHAR(50)
us_chunk_inx INT
us_props BLOB
us_size INT
us_sha1 VARCHAR(31)
us_mime VARCHAR(255)
us_media_type ENUM(...)
us_image_width INT
us_image_height INT
us_image_bits SMALLINT
Indexes

Interwiki

sites
site_id INT
site_global_key VARBINARY(32)
site_type VARBINARY(32)
site_group VARBINARY(32)
site_source VARBINARY(32)
site_language VARBINARY(32)

site_identifiers
si_site INT
si_type VARBINARY(32)
si_key VARBINARY(32)
Indexes

Caching tables

objectcache
keyname VARBINARY(255)
value MEDIUMBLOB
exptime DATETIME
Indexes

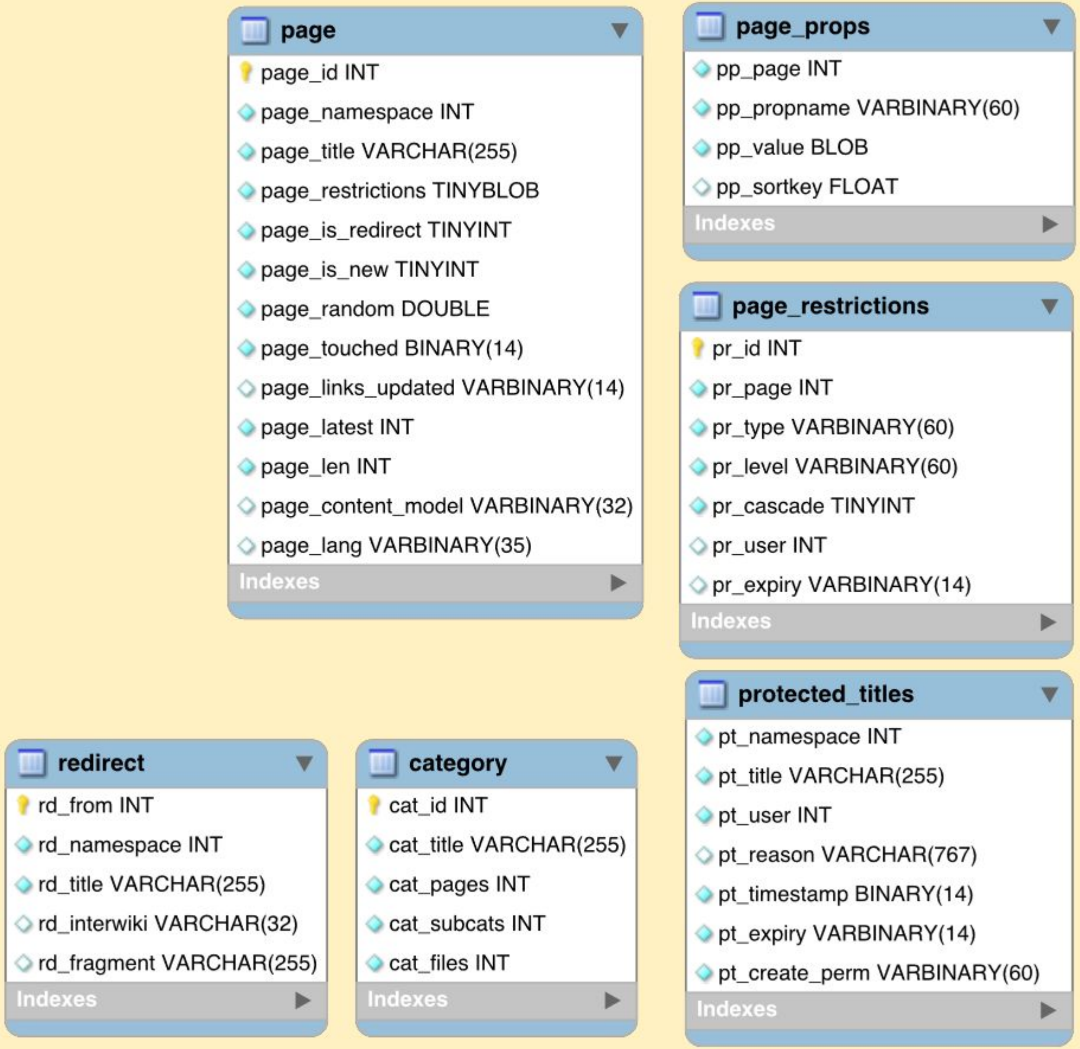
l10n_cache
lc_lang VARBINARY(32)
lc_key VARCHAR(255)
lc_value MEDIUMBLOB
Indexes

transcache
tc_url VARBINARY(255)
tc_contents TEXT
tc_time BINARY(14)
Indexes

Maintenance

updatelog
ul_key VARCHAR(255)
ul_value BLOB
Indexes

job
job_id INT
job_cmd VARBINARY(60)
job_namespace INT
job_title VARCHAR(255)
job_timestamp VARBINARY(14)
job_params BLOB



page

page_id INT
page_namespace INT
page_title VARCHAR(255)
page_restrictions TINYBLOB
page_is_redirect TINYINT
page_is_new TINYINT
page_random DOUBLE
page_touched BINARY(14)
page_links_updated VARBINARY(14)
page_latest INT
page_len INT
page_content_model VARBINARY(32)
page_lang VARBINARY(35)
Indexes

page_props

pp_page INT
pp_propname VARBINARY(60)
pp_value BLOB
pp_sortkey FLOAT
Indexes

page_restrictions

pr_id INT
pr_page INT
pr_type VARBINARY(60)
pr_level VARBINARY(60)
pr_cascade TINYINT
pr_user INT
pr_expiry VARBINARY(14)
Indexes

protected_titles

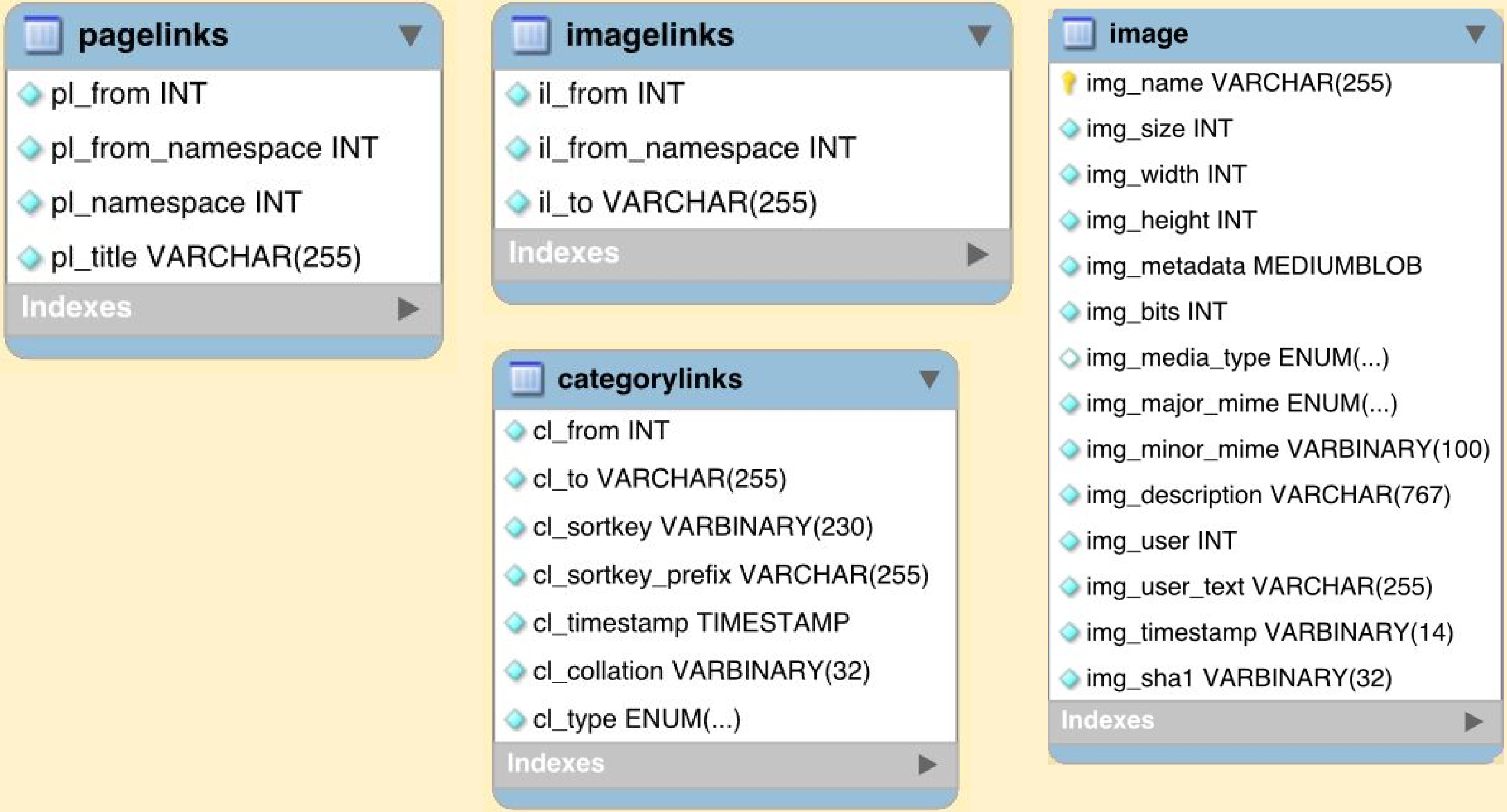
pt_namespace INT
pt_title VARCHAR(255)
pt_user INT
pt_reason VARCHAR(767)
pt_timestamp BINARY(14)
pt_expiry VARBINARY(14)
pt_create_perm VARBINARY(60)
Indexes

redirect

rd_from INT
rd_namespace INT
rd_title VARCHAR(255)
rd_interwiki VARCHAR(32)
rd_fragment VARCHAR(255)
Indexes

category

cat_id INT
cat_title VARCHAR(255)
cat_pages INT
cat_subcats INT
cat_files INT
Indexes



pagelinks

- pl_from INT
- pl_from_namespace INT
- pl_namespace INT
- pl_title VARCHAR(255)

Indexes

imagelinks

- il_from INT
- il_from_namespace INT
- il_to VARCHAR(255)

Indexes

categorylinks

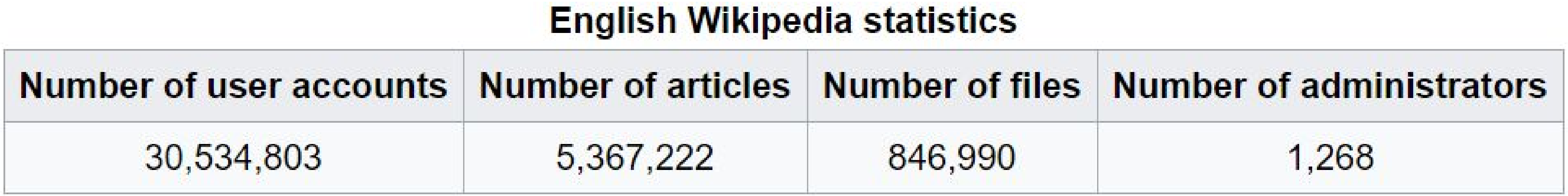
- cl_from INT
- cl_to VARCHAR(255)
- cl_sortkey VARBINARY(230)
- cl_sortkey_prefix VARCHAR(255)
- cl_timestamp TIMESTAMP
- cl_collation VARBINARY(32)
- cl_type ENUM(...)

Indexes

image

- img_name VARCHAR(255)
- img_size INT
- img_width INT
- img_height INT
- img_metadata MEDIUMBLOB
- img_bits INT
- img_media_type ENUM(...)
- img_major_mime ENUM(...)
- img_minor_mime VARBINARY(100)
- img_description VARCHAR(767)
- img_user INT
- img_user_text VARCHAR(255)
- img_timestamp VARBINARY(14)
- img_sha1 VARBINARY(32)

Indexes



English Wikipedia statistics

Number of user accounts	Number of articles	Number of files	Number of administrators
30,534,803	5,367,222	846,990	1,268

Uncompressed approximated data sizes:

- Article content ~ 50 GB
- User talk and data ~ 50 GB
- Full history of changes ~ 10 TB
- All files ~ 30 TB

Our local Wikipedia database

# of articles (without text)	# of files (names and infos)	# of categories
3,096,190	372,825	1,569,810

Our database size is just ~ 5 GB

```
mysql> select p.page_title, c.cl_to
    -> from page p use index( ),
    -> categorylinks c use index( )
    -> where p.page_id = c.cl_from and instr(c.cl_to, "War");

    5459 rows in set (12 min 23.30 sec)
```

```
+------------------------+---------------------------------+
| page_title             | cl_to                           |
+------------------------+---------------------------------+
| Achilles               | People_of_the_Trojan_War        |
| Achilles               | Thessalians_in_the_Trojan_War   |
| Abraham_Lincoln        | American_Civil_War              |
| Angolan_Armed_Forces   | Angolan_Civil_War               |
|                       [...]                             |
+------------------------+---------------------------------+
mysql> show index from page;

+-------+------------+---------------------------+-----------------+
| Table | Non_unique | Key_name                  | Column_name     |
+-------+------------+---------------------------+-----------------+
| page  |          0 | PRIMARY                   | page_id         |
| page  |          0 | name_title                | page_namespace  |
| page  |          0 | name_title                | page_title      |
| page  |          1 | page_random               | page_random     |
| page  |          1 | page_len                  | page_len        |
| page  |          1 | page_redirect_namespace_len | page_is_redirect |
| page  |          1 | page_redirect_namespace_len | page_namespace  |
| page  |          1 | page_redirect_namespace_len | page_len        |
+-------+------------+---------------------------+-----------------+
```

We want to make a query on the tables <page> and <categorylinks> filtering rows by means of the following columns:

1. page_id
2. cl_from
3. cl_to

```
mysql> show index from categorylinks;

+---------------+------------+------------------+---------------+
| Table         | Non_unique | Key_name         | Column_name   |
+---------------+------------+------------------+---------------+
| categorylinks |          0 | cl_from          | cl_from       |
| categorylinks |          0 | cl_from          | cl_to         |
| categorylinks |          1 | cl_timestamp     | cl_to         |
| categorylinks |          1 | cl_timestamp     | cl_timestamp  |
| categorylinks |          1 | cl_sortkey       | cl_to         |
| categorylinks |          1 | cl_sortkey       | cl_type       |
| categorylinks |          1 | cl_sortkey       | cl_sortkey    |
| categorylinks |          1 | cl_sortkey       | cl_from       |
| categorylinks |          1 | cl_collation_ext | cl_collation  |
| categorylinks |          1 | cl_collation_ext | cl_to         |
| categorylinks |          1 | cl_collation_ext | cl_type       |
| categorylinks |          1 | cl_collation_ext | cl_from       |
+---------------+------------+------------------+---------------+
```

1) <page_id> is a primary key which is always an index!

2 and 3) we are going to use a specific unique index.

```
CREATE UNIQUE INDEX cl_from
ON categorylinks (cl_from,cl_to);
```

```
mysql> select p.page_title, c.cl_to
    -> from page p use index(PRIMARY),
    -> categorylinks c use index(cl_from)
    -> where p.page_id = c.cl_from and instr(c.cl_to, "War");

     5459 rows in set (1.69 sec)


+-------------------------+---------------------------------+
| page_title              | cl_to                           |
+-------------------------+---------------------------------+
| Achilles                | People_of_the_Trojan_War        |
| Achilles                | Thessalians_in_the_Trojan_War   |
| Abraham_Lincoln         | American_Civil_War              |
| Angolan_Armed_Forces    | Angolan_Civil_War               |
|                      [...]                                |
+-------------------------+---------------------------------+
mysql> select p1.page_title, count(pl1.pl_title) as neigh_pages
    -> from pagelinks pl1, page p1
    -> where p1.page_id = pl1.pl_from
    -> group by p1.page_title
    -> having count(pl1.pl_title) = (
    ->   select max(v.neigh)
    ->   from (
    ->         select p2.page_title,count(pl2.pl_title) as neigh
    ->         from pagelinks pl2, page p2
    ->         where p2.page_id = pl2.pl_from
    ->         group by p2.page_title
    ->         ) v );
+-------------------------------------+-------------+
| page_title                          | neigh_pages |
+-------------------------------------+-------------+
| Beijing_Schmidt_CCD_Asteroid_Program |        1171 |
+-------------------------------------+-------------+
1 row in set (4 min 20.37 sec)
```

```
mysql> create view pageneighNoIndex(title,neigh) as
    -> select page_title,count(pl_title) as neigh_pages
    -> from pagelinks, page
    -> where page_id = pl_from
    -> group by page_title;
Query OK, 0 rows affected (0.05 sec)

mysql> select pn1.title, pn1.neigh
    -> from pageneighNoIndex pn1
    -> where pn1.neigh = (select max(pn2.neigh)
    ->   from pageneighNoIndex pn2);
+--------------------------------------+-------+
| title                                | neigh |
+--------------------------------------+-------+
| Beijing_Schmidt_CCD_Asteroid_Program |  1171 |
+--------------------------------------+-------+
1 row in set (3 min 36.65 sec)
```

```
mysql> create view pageneigh(title,neigh) as
    -> select page_title,count(pl_title) as neigh_pages
    -> from pagelinks use index(pl_from),
    -> page use index(PRIMARY)
    -> where page_id = pl_from
    -> group by page_title;
Query OK, 0 rows affected (0.08 sec)

mysql> select pn1.title, pn1.neigh
    -> from pageneigh pn1
    -> where pn1.neigh = (select max(pn2.neigh)
    ->   from pageneigh pn2);
+--------------------------------------+-------+
| title                                | neigh |
+--------------------------------------+-------+
| Beijing_Schmidt_CCD_Asteroid_Program |  1171 |
+--------------------------------------+-------+
1 row in set (40.26 sec)
```

```
mysql> select page_title,count(pl_title) as neigh_pages
    -> from pagelinks use index(pl_from),
    -> page use index(PRIMARY)
    -> where page_id = pl_from
    -> group by page_title
    -> order by neigh_pages DESC
    -> limit 1;
+--------------------------------------+-------------+
| page_title                           | neigh_pages |
+--------------------------------------+-------------+
| Beijing_Schmidt_CCD_Asteroid_Program |        1171 |
+--------------------------------------+-------------+
1 row in set (22.89 sec)
```

```
mysql> create view biggest_img_per_user_2017_noIndex as
    -> select img_user_text, max(img_size) as biggest_img
    -> from image
    -> where img_timestamp > 20170000000000
    -> group by img_user_text;
Query OK, 0 rows affected (0.02 sec)

mysql> select biggest_img
    -> from biggest_img_per_user_2017_noIndex
    -> where img_user_text = "Theo's Little Bot";
+-------------+
| biggest_img |
+-------------+
|     1879744 |
+-------------+
1 row in set (3 min 34.19 sec)
```

```
mysql> create unique index Covering_Index
    -> on image (img_timestamp,img_user_text,img_size);
Query OK, 0 rows affected (11.51 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> create view biggest_img_per_user_2017 as
    -> select img_user_text, max(img_size) as biggest_img
    -> from image use index(Covering_Index)
    -> where img_timestamp > 20170000000000
    -> group by img_user_text;
Query OK, 0 rows affected (0.05 sec)

mysql> select biggest_img
    -> from biggest_img_per_user_2017
    -> where img_user_text = "Theo's Little Bot";
+-------------+
| biggest_img |
+-------------+
|     1879744 |
+-------------+
1 row in set (0.77 sec)
```