

Bases de datos* en Wikipedia

Jaime Crespo, Sr Database Administrator



WIKIMEDIA
FOUNDATION

***relacionales**

Introducción



WIKIMEDIA
FOUNDATION

¿Quién es Jaime Crespo?

- Empleado de la Wikimedia Foundation desde hace 5 años
- Persistence Team (1 of 3), Site Reliability Engineering
- Equipo encargado de bases de datos y backups



¿Qué haces?

- Mantenimiento de la infraestructura de bases de datos y backups
 - Monitorización de salud de servicios
 - Respuesta a incidentes
- Compra y setup de equipos
- Automatización de tareas
- Consultoría en temas de rendimiento y seguridad
 - Optimización de código y consultas SQL
 - Despliegue de cambios de esquema

¿Cómo has terminado aquí?

- Miembro fundador de Púlsar, Linux geek
- Prácticas de empresa en BIFI
- Oficina de Software Libre de UZ
- Instructor/Desarrollador @Warp Networks
- Instructor/Consultor MySQL @Percona
- Consultor independiente

Sobre esta charla

¡Disclaimer! Esto refleja solo nuestra experiencia, que puede variar mucho en otros entornos

- Enfocada a estudiantes de grado de la asignatura de BBDD
- Nuestro entorno:
 - Software libre
 - Web
 - OLTP
 - “Grandes” clusters
 - Metal
 - Maximizar rendimiento/\$

Entorno de producción: Cómo funciona Wikipedia por dentro (incluidas las bases de datos)



WIKIMEDIA
FOUNDATION

Algunos datos sobre nuestros proyectos

- Wikipedia: top 5 de páginas más visitadas del mundo
 - [26 mil millones de páginas vistas](#) en abril de 2020
 - Entre 100-200K “GET”s por minuto
- 49 millones de [ediciones](#) en Marzo de 2020
- [>300 idiomas](#) / [>900 wikis](#)
- [>6 millones de páginas wiki en inglés](#)
- [>60 millones de archivos libres en Commons](#) (250 TB)
- [85 millones de “items” en Wikidata](#)

Algunos datos de las bases de datos

- ~600 TB de datos en bases de datos relacionales
- [220 instancias de MariaDB 10.1/10.4](#)
- 24+ grupos de replicación
- [400 - 600 mil Consultas SQL por segundo](#)

Arquitectura física

- +1400 servidores físicos
- 5 centros de datos, 2 primarios

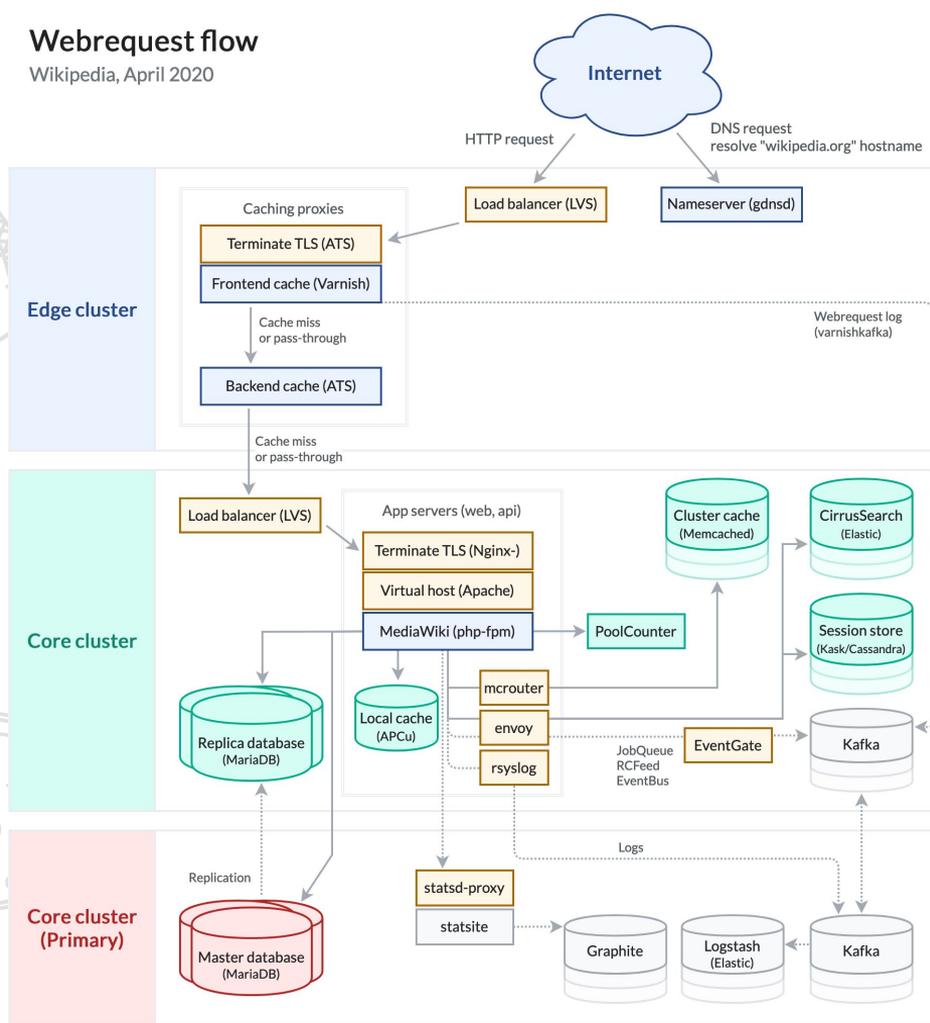


Pila de software

- **Debian Linux** Stretch/Buster (os)
- Linux Virtual Server (load balancing)
- ATS (cache)
- **Mediawiki** on top of PHP7 (application servers)
- Memcache
- Kafka (jobqueue, messaging)
- Elasticsearch (indexing & search)
- **MariaDB** 10.1/10.4 (data)
- OpenStack Swift (media)

Webrequest flow

Wikipedia, April 2020



Legend: Blocking connect Async or post-send Storage Proxy Responds to user Responds to MW

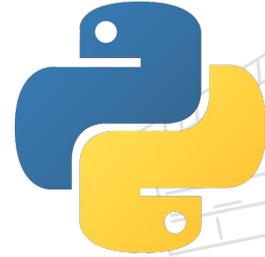
Otro software

- Puppet (configuration management)
- Git (code versioning)
- Hadoop (analytics)
- Cassandra (sessions/APIs)
- Jenkins (CI)
- K8s (orchestrator)
- Icinga + Prometheus/Grafana (monitoring)
- HAProxy (proxy)
- Gdnsd (DNS)

Lenguajes de programación

- ¡TODOS!
 - PHP
 - SQL
 - Java
 - Js/Node
 - Python
 - Ruby
 - Bash
 - Lua
 - SPARQL
 - C/C++

...



BASH
THE BOURNE-AGAIN SHELL

Principios (I)

- Alta disponibilidad mediante redundancia a todos los niveles “shared nothing”:
 - red, servidores, balanceadores de carga, discos (RAID1 o 10), centros de datos, proveedores, sres
- “Infrastructure as code”:

<https://phabricator.wikimedia.org/source/operations-puppet/history/production/>

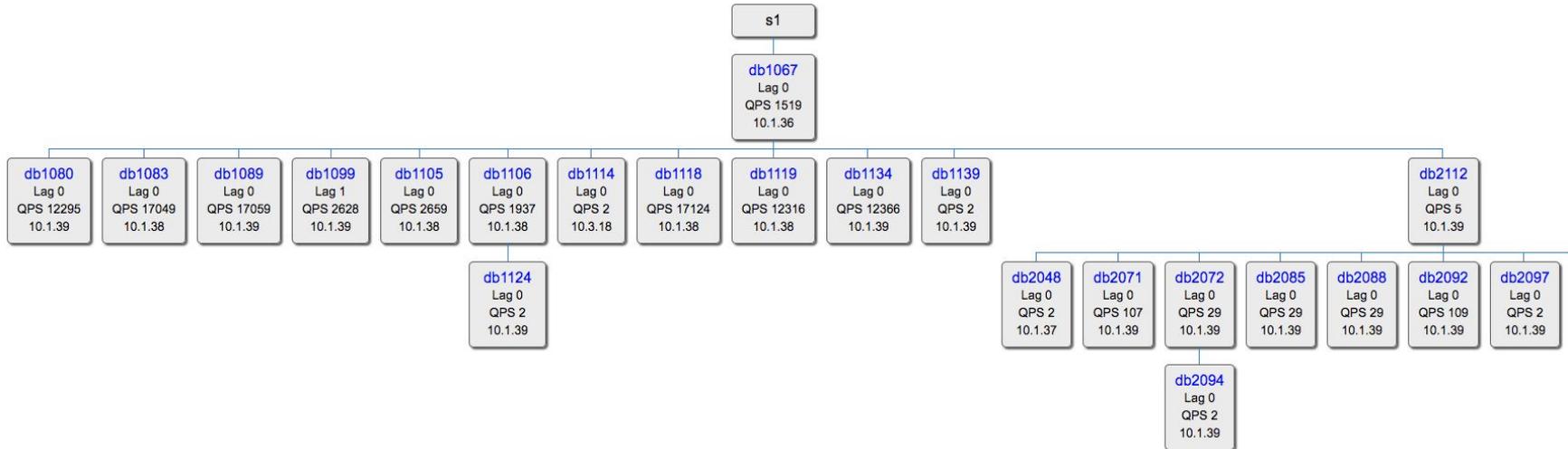
Principios (II)

- Solo uso de software de código abierto
- Todo en metal auto-hosteado (aunque tenemos varios servicios virtualizados)
- En pleno proceso de migrar un diseño monolítico (Mediawiki) a microservicios (Kubernetes)
- Continuous Integration/Delivery (docenas de deployments al día a producción)

Entorno de bases de datos

- 2 DCs con bases de datos, activo-pasivo
- Replicación con maestros intermedios

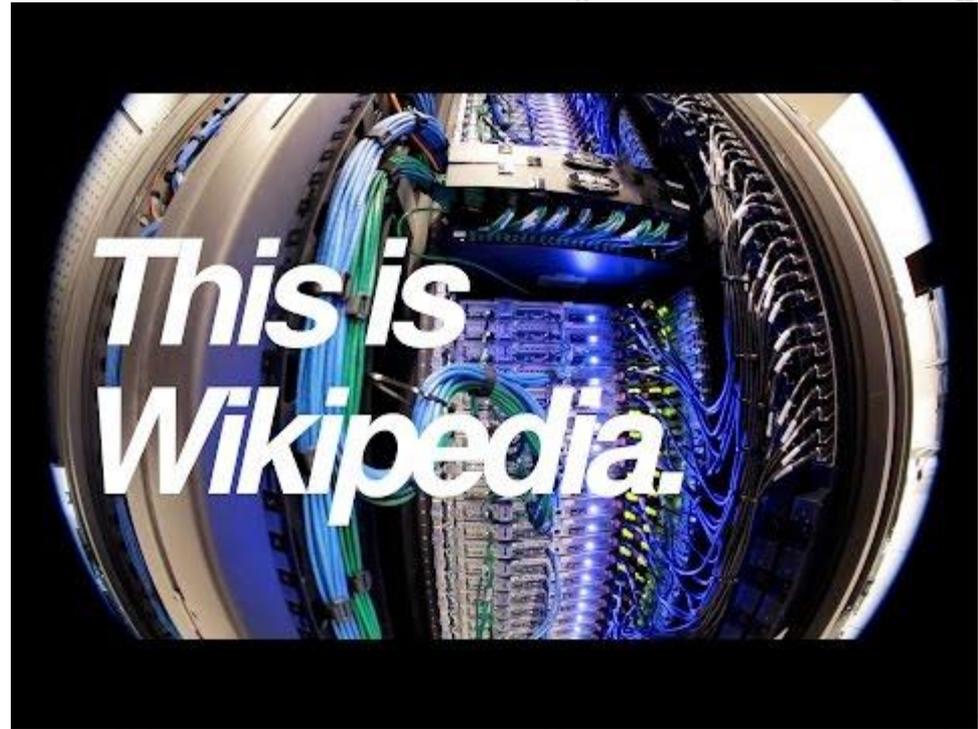
<https://dbtree.wikimedia.org/>



Hardware & configuración BD

- Servidores x86_64 (2P / 8H)
- 512 GB of RAM
- 4TB - SSDs en RAID10 (HW)
- BBDD entre 90GB y 12TB)
- 1 servidor puede realizar cómodamente 20K QPS
- Configuración MariaDB:

https://phabricator.wikimedia.org/source/operations-puppet/browse/production/modules/role/templates/mariadb/mysql_d_config/production.my.cnf.erb?blame=off



¿Para qué usamos MariaDB?

- Datos relacionales principales (metadatos de usuarios, texto y ficheros, ...)
 - Servir peticiones de navegadores
 - API de edición y consulta
- Almacenamiento clave-valor seguro:
 - Contenido de cada página (revisiones)
- Caché basada en disco:
 - Nivel secundario de caché para wikitexto parseado, etc.
- Estadísticas (obsoleto por Hadoop y otros)
- Mayoría de servicios internos con necesidades de bases de datos

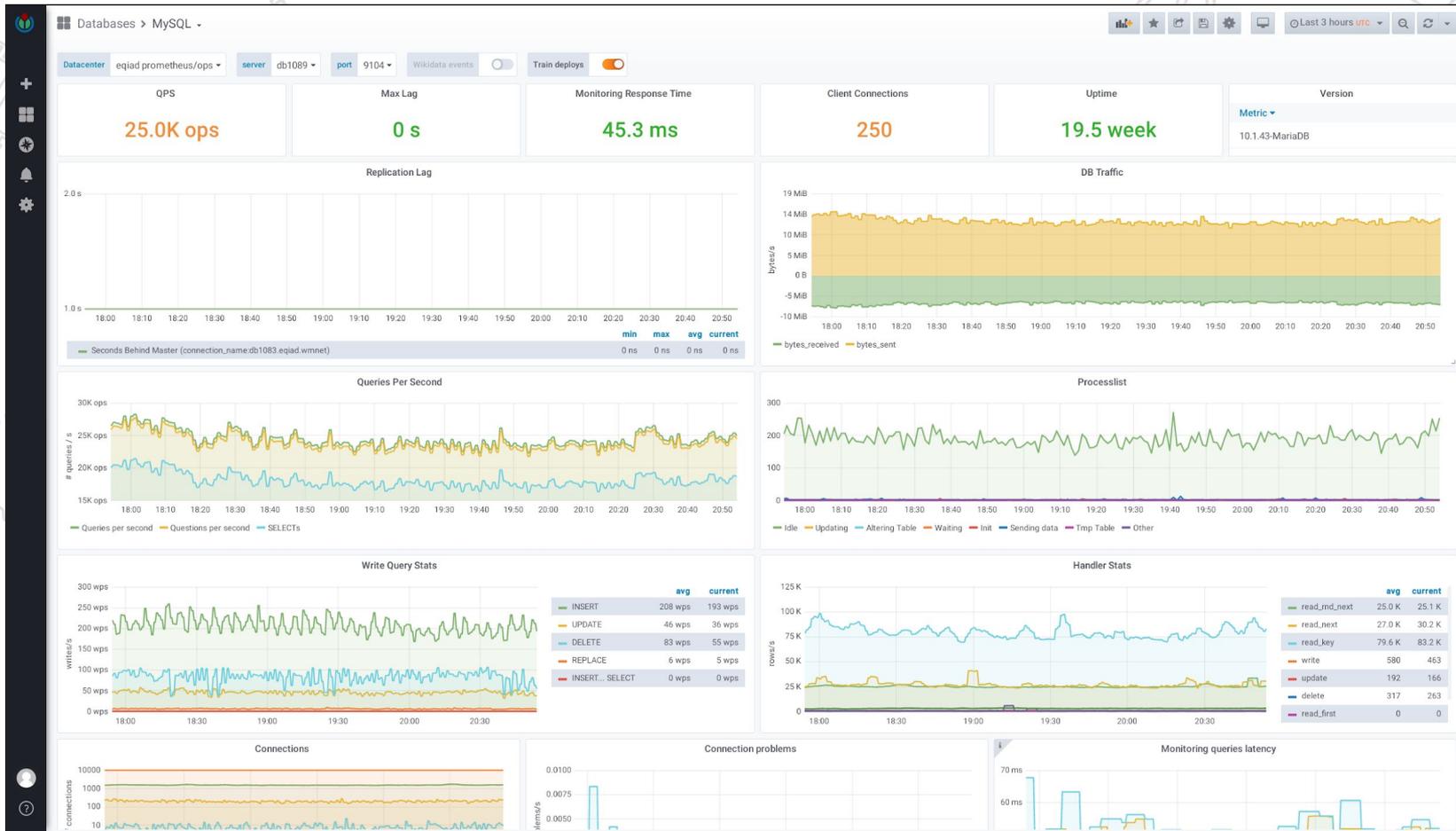
¿Para qué no usamos MariaDB?

- Restful API y sesiones:
 - Cassandra
- Procesado de estadísticas
 - Hadoop et al.
- Memory caching
 - Memcache
- Cola de trabajo:
 - Kafka
- Búsqueda y logs:
 - ELK
- Datos geográficos
 - PostGIS

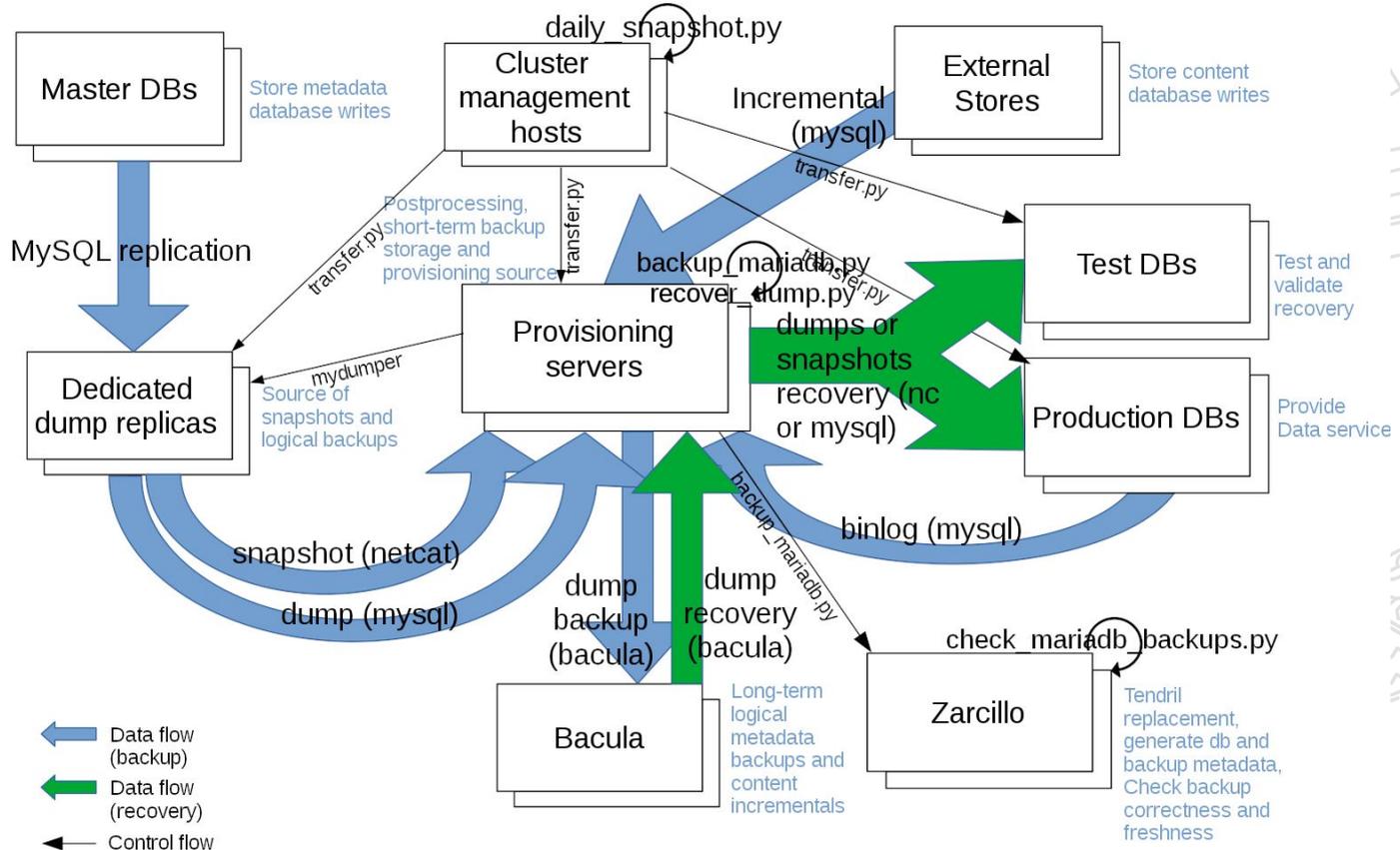
Transparencia para Mena de por qué Oracle no funcionaría

- Coste: \$100 millones en licencias
- + Coste de migración, training
- Imposibilidad de arreglo de bugs/parches propios
- Incompatibilidad de licencia y otras restricciones
- Facilidad de la replicación/shared nothing
- Poco peso del servicio
- Facilidad de actualización
- Popularidad y facilidad de uso por desarrolladores

Monitorización



Copias de seguridad



Caso práctico #1: Diseñando un esquema para Wikipedia



WIKIMEDIA
FOUNDATION

Dinámica de los casos prácticos

- Se presenta un problema real de diseño de bases de datos
- Discusión breve (5 minutos)
- Analizamos las propuestas y comparamos las ventajas y desventajas de cada uno, y luego lo comparamos con el que se usa realmente

Caso #1: Diseñar un esquema para Wikipedia

- ¿Qué entidades necesitaríamos?
- ¿Qué relaciones?
- ¿Cuáles serían las consultas más frecuentes?
- ¿Cuáles crees que son los principales problemas de escalado?

Entidades potenciales relacionadas con contenido

- **Página:** ¿Necesitamos una entidad así?
- **Edición:** ¿Diferente de página?
- **Diff:** ¿Debería ser una entidad “fuerte”?
- **Revisión:** ¿Resultaría luego en una tabla muy grande?

¿Cómo resolvemos los tipos de página y sus propiedades?

- **Páginas de discusión:** ¿Misma entidad o diferente?
- **Categorías, imágenes (archivos), redirecciones:** ¿Son páginas normales o deberían ser entidades independientes?
- Misma pregunta para **páginas de descripción de imágenes**
- **Categorías para páginas:** ¿Cómo implementarlas?

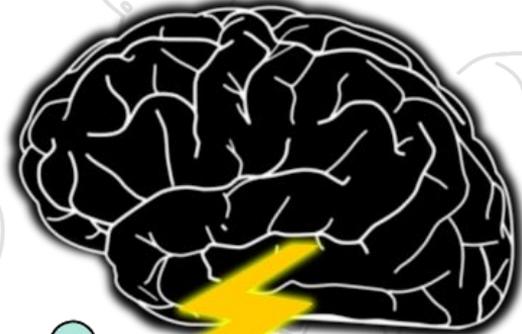
¿Usuarios y otras propiedades?

- **Usuarios:** ¿Qué guardamos y cómo los identificamos?
- **Protecciones:** Algunas páginas tienen restricciones sobre quién puede editarlas
- **Other propiedades** (¿etiquetas?)

Identificadores

- ¿Cómo solucionar el problema de **2 páginas que deberían tener el mismo título**: El rey león (1994) y El rey león (2019)?
- ¿Deberíamos usar **el nombre de la página o un identificador único** para cada página?
- ¿Y para **identificadores de revisiones**?
- Si necesitáramos uno, ¿debería ser un **UUID o un identificador numérico**?

Brainstorming time



User

user
user_id INT
user_name VARCHAR(255)
user_real_name VARCHAR(255)
user_password TINYBLOB
user_newpassword TINYBLOB
user_newpass_time BINARY(14)
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)

user_properties
up_user INT
up_property VARBINARY(255)
up_value BLOB

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

actor
actor_id BIGINT
actor_user INT
actor_name VARCHAR(255)

bot_passwords
bp_user INT
bp_app_id VARBINARY(32)
bp_password TINYBLOB
bp_token BINARY(32)
bp_restrictions BLOB
bp_grants BLOB

Permissions

user_groups
ug_user INT
ug_group VARBINARY(255)
ug_expiry VARBINARY(14)

user_former_groups
ufg_user INT
ufg_group VARBINARY(255)

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)

protected titles
pt_namespace INT
pt_title VARCHAR(255)
pt_user INT
pt_reason_id BIGINT
pt_timestamp BINARY(14)
pt_expiry VARBINARY(14)
pt_create_perm VARBINARY(60)

ipblocks
ipb_id INT
ipb_address TINYBLOB
ipb_user INT
ipb_by_actor BIGINT
ipb_reason_id BIGINT
ipb_timestamp BINARY(14)
ipb_auto BOOL
ipb_anon_only BOOL
ipb_create_account BOOL
ipb_enable_autoblock BOOL
ipb_expiry VARBINARY(14)
ipb_range_start TINYBLOB
ipb_range_end TINYBLOB
ipb_deleted BOOL
ipb_block_email BOOL
ipb_allow_usertalk BOOL
ipb_parent_block_id INT
ipb_sitewide BOOL

ipblocks_restrictions
ir_ipb_id INT
ir_type TINYINT(1)
ir_value INT

Logging

logging
log_id INT
log_type VARBINARY(32)
log_action VARBINARY(32)
log_timestamp BINARY(14)
log_actor BIGINT
log_namespace INT
log_title VARCHAR(255)
log_page INT
log_comment_id BIGINT
log_params BLOB
log_deleted TINYINT

log_search
ls_field VARBINARY(32)
ls_value VARCHAR(255)
ls_log_id INT

comment
comment_id BIGINT
comment_hash INT
comment_text BLOB
comment_data BLOB

Tags

change_tag
ct_id INT
ct_rc_id INT
ct_log_id INT
ct_rev_id INT
ct_params BLOB
ct_tag_id INT

change_tag_def
ctd_id INT
ctd_name VARBINARY(255)
ctd_user_defined TINYINT(1)
ctd_count BIGINT

Recent changes

recentchanges
rc_id INT
rc_timestamp VARBINARY(14)
rc_actor BIGINT
rc_namespace INT
rc_title VARCHAR(255)
rc_comment_id BIGINT
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

watchlist
wl_id INT
wl_user INT
wl_namespace INT
wl_title VARCHAR(255)
wl_notificationtimestamp VARBINARY(14)

Pages

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

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

archive
ar_id INT
ar_namespace INT
ar_title VARCHAR(255)
ar_comment_id BIGINT
ar_actor BIGINT
ar_timestamp BINARY(14)
ar_minor_edit TINYINT
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)

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

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

Revisions

revision
rev_id INT
rev_page INT
rev_text_id INT
rev_comment VARCHAR(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 VARCHAR(32)
rev_content_model VARCHAR(32)
rev_content_format VARCHAR(64)

slots
slot_revision_id BIGINT
slot_content_size INT
slot_role_id BIGINT
slot_content_id BIGINT
slot_origin BIGINT

slot_roles
role_id SMALLINT
role_name VARCHAR(64)

ip_changes
ipc_rev_id INT
ipc_rev_timestamp BINARY(14)
ipc_hex VARCHAR(35)

content
content_id BIGINT
content_size INT
content_sha1 VARBINARY(32)
content_model SMALLINT
content_address VARBINARY(255)

content_models
model_id SMALLINT
model_name VARCHAR(64)

text
old_id INT
old_text MEDIUMBLOB
old_flags TINYBLOB

Link tables

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

iwlinks
iw_from INT
iw_prefix VARBINARY(20)
iw_title VARCHAR(255)

externallinks
el_id INT
el_from INT
el_to BLOB
el_index BLOB
el_index_60 VARBINARY(60)

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

imagelinks
il_from INT
il_from_namespace INT
il_to VARCHAR(255)

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

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(...)

Statistics

site_stats
ss_row_id INT
ss_total_edits BIGINT
ss_good_articles BIGINT
ss_total_pages BIGINT
ss_users BIGINT
ss_active_users BIGINT
ss_images BIGINT

Search

searchindex
si_page INT
si_title VARCHAR(255)
si_text MEDIUMTEXT

Maintenance

job
job_id INT
job_cmd VARBINARY(60)
job_namespace INT
job_title VARCHAR(255)
job_timestamp VARBINARY(14)
job_params MEDIUMBLOB
job_random INTEGER
job_attempts INTEGER
job_token VARBINARY(32)
job_token_timestamp VARBINARY(14)
job_sha1 VARBINARY(32)

updatelog
ul_key VARCHAR(255)
ul_value BLOB

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_id BIGINT
img_actor BIGINT
img_timestamp VARBINARY(14)
img_sha1 VARBINARY(32)

oldimage
oi_name VARCHAR(255)
oi_archive_name VARCHAR(255)
oi_size INT
oi_width INT
oi_height INT
oi_bits INT
oi_description_id BIGINT
oi_actor BIGINT
oi_timestamp BINARY(14)
oi_metadata MEDIUMBLOB
oi_media_type ENUM(...)
oi_major_mime ENUM(...)
oi_minor_mime VARBINARY(100)
oi_deleted TINYINT
oi_sha1 VARBINARY(32)

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_id BIGINT
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_id BIGINT
fa_actor BIGINT
fa_timestamp BINARY(14)
fa_deleted TINYINT
fa_sha1 VARBINARY(32)

uploadstash
us_user INT
us_key VARBINARY(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

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_protocol VARBINARY(32)
site_domain VARCHAR(255)
site_data BLOB
site_forward BOOL
site_config BLOB

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

Interwiki
iw_prefix VARCHAR(32)
iw_url BLOB
iw_api BLOB
iw_wikiid VARCHAR(64)
iw_local BOOL
iw_trans TINYINT

Caching tables

querycache
qc_type VARBINARY(32)
qc_value INT
qc_namespace INT
qc_title VARCHAR(255)

querycachetwo
qcc_type VARBINARY(32)
qcc_value INT
qcc_namespace INT
qcc_title VARCHAR(255)
qcc_namespacewo INT
qcc_titletwo VARCHAR(255)

querycache_info
qci_type VARBINARY(32)
qci_timestamp BINARY(14)

l10n_cache
lc_lang VARBINARY(32)
lc_key VARCHAR(255)

ResourceLoader

module_deps
md_module VARBINARY(255)
md_skin VARBINARY(32)
md_deps MEDIUMBLOB

Disclaimers

- La mejor solución en papel no es necesariamente la mejor para producción
 - Podría ser muy difícil migrar la lógica previa (aplicación usando el mismo esquema durante casi 20 años) o no merecer la pena
 - El rendimiento no es la única métrica: seguridad, escalabilidad, estabilidad, simplicidad, etc.
- Hay muchos ejemplos de compromisos así en nuestra aplicación

Descripción del caso #2

← https://en.wikipedia.org/wiki/Main_Page



The image shows a screenshot of the Wikipedia Main Page. On the left side, there is a navigation menu with the following items: Main page, Contents, Featured content, Current events, Random article, Donate to Wikipedia, Wikipedia store, Interaction (Help, About Wikipedia, Community portal, Recent changes, Contact page), and Tools (What links here, Related changes, Upload file, Special pages, Permanent link, Page information, Wikidata item). A large red arrow points from the 'Random article' link to the 'From today's featured article' section on the right. The featured article is titled 'Brill Tramway locomotive' and includes a small image of a red steam locomotive.

- A la izquierda de cada página hay un enlace a “Página aleatoria”
- Debe permitir filtrar por “espacio de nombres” (p.ej.: no todas las páginas son artículos)
- Es bastante importante, comparado con “voy a tener suerte” de Google, ya que tiene que dar una idea de los artículos a encontrar en una Wiki

Restricciones del caso #2



- Tiene que funcionar igual de rápido que una página normal, pero por razones obvias los resultados no se pueden cachear
- Tiene que ser creíblemente pseudoaleatoria
- Siempre tiene que devolver un resultado
- Tiene que funcionar siempre, independientemente del número de páginas, y escalar de 1 a millones de páginas

Potenciales soluciones

- ORDER BY rand() LIMIT 1
- Usar un entero bien distribuido y seleccionar uno al azar
- ¿Preguntas?
 - ¿Qué índices serían beneficiosos en cada caso?
 - ¿Cómo consultar el número total de ids?
 - ¿Cómo gestionar borrados?



Solución real: Diseño de la base de datos

(I)

```
CREATE TABLE /*_*/page (
```

```
[...]
```

```
-- A page name is broken into a namespace and a title.
```

```
-- The namespace keys are UI-language-independent constants,
```

```
-- defined in includes/Defines.php
```

```
page_namespace int NOT NULL,
```

```
-- The rest of the title, as text.
```

```
-- Spaces are transformed into underscores in title storage.
```

```
page_title varchar(255) binary NOT NULL,
```

```
-- 1 indicates the article is a redirect.
```

```
page_is_redirect tinyint unsigned NOT NULL default 0,
```

```
[...]
```

```
-- Random value between 0 and 1, used for Special:Randompage
```

```
page_random real unsigned NOT NULL,
```

Solución real: Diseño de la base de datos (II)

[...]

```
CREATE INDEX /*i*/page_random ON /*_*/page (page_random);
```

Solución real: Código

```
protected function getQueryInfo( $randstr ) {
    $redirect = $this->isRedirect() ? 1 : 0;
    $tables = [ 'page' ];
    $conds = array_merge( [
        'page_namespace' => $this->namespaces,
        'page_is_redirect' => $redirect,
        'page_random >= ' . $randstr
    ], $this->extra );
    $joinConds = [];

    // Allow extensions to modify the query
    Hooks::run( 'RandomPageQuery', [ &$tables, &$conds, &$joinConds ] );

    return [
        'tables' => $tables,
        'fields' => [ 'page_title', 'page_namespace' ],
        'conds' => $conds,
        'options' => [
            'ORDER BY' => 'page_random',
            'LIMIT' => 1,
        ],
        'join_conds' => $joinConds
    ];
}
```

From: [mediawiki/core/includes/specials/SpecialRandompage.php](https://www.mediawiki.org/wiki/mediawiki/core/includes/specials/SpecialRandompage.php)

Solución real: Consulta generada

```
SELECT    page_title, page_namespace
FROM      page
LEFT JOIN page_props
ON        page_id = pp_page AND
          pp_propname = ?
WHERE     page_namespace IN (...) AND
          page_is_redirect = 0 AND
          page_random >= $rand
ORDER BY  page_random
LIMIT    1;
```

Solución real: Rendimiento (I)

```
mysql> EXPLAIN SELECT ... \G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: page
         type: range
possible_keys: name_title,page_random,page_redirect_namespace_len
           key: page_random
        key_len: 8
           ref: NULL
         rows: 20473233
       Extra: Using where
***** 2. row *****
      id: 1
  select_type: SIMPLE
        table: page_props
         type: eq_ref
possible_keys: PRIMARY,pp_propname_page,pp_propname_sortkey_page
           key: PRIMARY
        key_len: 66
           ref: enwiki.page.page_id,const
         rows: 1
       Extra: Using where; Using index; Not exists
```

Solución real: Rendimiento (II)

```
mysql> SELECT ... FROM sys.x$statement_analysis ...
```

```
***** 1. row *****
```

```
exec_count: 27126203
```

```
max_latency: 450802755000
```

```
avg_latency: 755698000
```

```
lock_latency: 2224515688000000
```

```
rows_sent: 27125869
```

```
rows_sent_avg: 1
```

```
rows_examined: 0
```

```
rows_examined_avg: 0
```

```
rows_affected: 0
```

```
rows_affected_avg: 0
```

```
tmp_tables: 0
```

```
tmp_disk_tables: 0
```

```
rows_sorted: 777598
```

```
sort_merge_passes: 0
```

```
+-----+  
| sys.format_time(avg_latency) |  
+-----+  
| 755.70 us |  
+-----+
```


Cambios de mentalidad más importantes trabajando “a escala”

- Programación/automatización de infraestructura
- Trabajo en equipo/saber dónde ceder (y dónde no)
- Paciencia en aplicar cambios
- La teoría mola y (a veces) muy útil (estadística, estructura de datos, ...)
- Aunque no os dediquéis a las BBDD, trabajaréis con ellas

Para saber más...

- <https://www.slideshare.net/jynus/mysql-schema-design-in-practice>
- <https://www.usenix.org/conference/srecon19emea/presentation/mouzeli> (video)
- <https://www.slideshare.net/jynus/wikipedia-burgos-devfest-2017>
- <https://www.slideshare.net/jynus/backing-up-wikipedia-databases>
- <https://wikitech.wikimedia.org/wiki/Portal:Wikitech>

¡Puedes ayudarnos!

- Ofertas de trabajo:
<https://wikimediafoundation.org/about/jobs/>
- GSoC
- “Becas” de desarrollo a contribuidores
- Voluntariado

¡GRACIAS!

Estoy en:

jcrespo@wikimedia.org



WIKIMEDIA
FOUNDATION