Database Concepts for a new generation of CMS

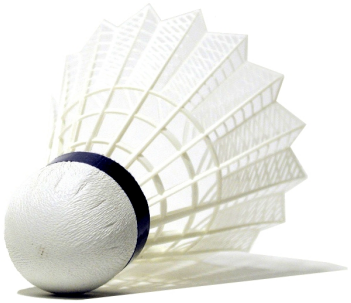
Drupal - eZPublish - ...

Thomas Koch

Young Media Concepts GmbH

July 31, 2008

IT mögen.



Outline

What this talk is about

How to use a relational database effectively for a content management system.

What this talk is not about

- Alternative storage systems (CouchDB, XML)
- Other components of a CMS then storage

What does a CMS?

- Store content (Content = highly structured, mostly textual informations)
- Organize content
- Organize content relations
- Present content in different ways
- Implement Workflows, Permissions
- Search content

Examples of CMS

- CMS: Drupal, eZPublish, Plone
- Groupware: eGroupWare, Horde, Exchange
- Issue Tracker: Flyspray, OTRS, Bugzilla
- ERP: Organize documents: and workflows

What are not CMS's?

- OpenOffice, GIMP: Editors
- Gallery2: Organizes Images, not textual content
- Mailreaders, RSS reader, Browser: retrieve and present
- Not many Webapplications that are not CMS!
- more . . . ?

Outline

Tables are static

- Drupal: node, node_revisions, comments, poll, book, ...
- eGroupWare: egw_addressbook, egw_addressbook_extra, egw_cal, egw_cal_extra, ...
- OTRS: ticket, service, faq_item, ...

New modules come with new table definitions.

Schema for a Drupal module

```
function subscriptions_schema() {

  $schema['subscriptions'] = array(
    'fields'          => array(
      'sid'           => array('type' => 'serial', 'unsigned' => TRUE,
                                 'not null' => TRUE, 'disp-width' => '10'),
      'module'        => array('type' => 'varchar', 'length' => '64',
                                 'not null' => FALSE),
      'field'         => array('type' => 'varchar', 'length' => '32',
                                 'not null' => FALSE),
      'value'         => array('type' => 'varchar', 'length' => '237',
                                 'not null' => FALSE),
      ...more fields
    'primary key'     => array('sid'),
    'indexes'         => array(
      'module'        => array('module', 'field', 'value'),
      'recipient_uid' => array('recipient_uid')),
  );
```

Schema for an eGroupWare module

```
$phpgw_baseline = array(
 'egw_bookmarks' => array(
  'fd' => array(
   'bm_id'=>array('type'=>'auto','nullable'=>False),
   'bm_owner'=>array('type'=>'int','precision'=>4,'nullable'=>True),
   'bm_access'=>array('type'=>'varchar','precision'=>255,'nullable'=>True),
   'bm_url'=>array('type'=>'varchar','precision'=>255,'nullable'=>True),
   'bm_name'=>array('type'=>'varchar','precision'=>255,'nullable'=>True),
   'bm_desc'=>array('type'=>'text','nullable'=>True),
   'bm_keywords'=>array('type'=>'varchar','precision'=>255,'nullable'=>True),
   'bm_category'=>array('type'=>'int','precision'=>4,'nullable'=>True),
   'bm_rating'=>array('type'=>'int','precision'=>4,'nullable'=>True),
   'bm_info'=>array('type'=>'varchar','precision'=>255,'nullable'=>True),
   'bm_visits'=>array('type'=>'int','precision'=>4,'nullable'=>True)
  ),
  'pk' => array('bm_id'),
  'fk' => array(),
  'ix' => array(),
  'uc' => array()
 )
);
```

conclusion

Modifying the database scheme in such systems requires programming and is a major surgery.

Outline

columns in egw_addressbook

- contact_id
- contact_tid
- contact_owner
- contact_private
- cat_id
- n_family
- n_given
- n_middle
- n_prefix
- n_suffix
- n_fn
- n_fileas
- contact_bday
- org_name
- org_unit
- contact_title
- contact_role
- contact_assistent
- contact_room
- adr_one_street
- adr_one_locality
- adr_one_region
- adr_one_postalcode
- adr_one_countryname
- contact_label
- adr_two_street
- adr_two_street2
- adr_two_locality
- adr_two_region
- adr_two_postalcode
- adr_two_countryname
- tel_work
- tel_cell
- tel_fax
- tel_assistent
- tel_car
- tel_pager
- tel_home
- tel_fax_home
- tel_cell_private
- tel_prefer
- contact_email
- contact_email_home
- contact_url
- contact_url_home
- contact_freebusy_uri
- contact_calendar_uri
- contact_note
- contact_tz
- contact_geo
- contact_pubkey
- contact_created
- contact_creator
- contact_modified
- contact_modifier
- contact_jpegphoto
- account_id
- contact_etag

problems with static tables

- only two web addresses possible
- who needs tel_car and tel_pager anymore?
- No field for a client id

custom fields in eGroupWare

Table egw_addressbook_extra

- contact_id
- contact_owner
- contact_name
- contact_value

This is a first step in the direction of the *Entity-Attribute-Value* Pattern in database design.[a]

[a] http://en.wikipedia.org/wiki/Entity-Attribute-Value_model

Entity-Attribute-Value Pattern

Entity	Attribute	Value
1	n_family	Koch
1	n_given	Thomas
2	n_family	Musterman
2	n_given	Max
2	n_prefix	Dr.

eZ Publish uses this pattern in a more advanced version.

EAV Pattern - Advantages

- Flexibility

EAV Pattern - Disadvantages

- Many slow JOINs necessary
- Everything is a varchar
- Specialised tools necessary to browse the DB
- Hand coding SQL for this pattern is a pain
- Many build-in database features aren't available
  - referential integrity checks
  - domain integrity checks
  - indexes

Solution: Ask Lukas Kahwe Smith

- LAMP software architect Liip since 2008
- database: MySQL, SQLite, Oracle, PostGreSQL
- MySQL 5.0 Certified Developer
- Publications, Speeches and Contributions in the field of databases
- `www.pooteeweet.org`

Lukas K. Smith

Presentation: SQL (un)patterns

Split up into separate tables with as many columns as necessary, create DDL on the fly if necessary

One Table for each content class

content_object_version_myclass1

+content_object_id: integer
+version_nr: integer
+locale: string
+name: string
+published: integer
+creator_id: string
+created: timestamp
+additional fields from attributes: mixed
+attrib1_weisnich: integer
+attrib1_blablu: string
+attrib2_sonst

content_object_version_myclass2

+content_object_id: integer
+version_nr: integer
+locale: string
+name: string
+published: integer
+creator_id: string
+created: timestamp
+additional fields from attributes: mixed
+attrib1_datafield1: integer
+attrib1_datafield2: string
+attrib2_datafield: blob
+attrib3_datafield: timestamp

DDL on the fly

How it works

- Adding an attribute: ALTER TABLE ADD COLUMN
- Deleting an attribute: ALTER TABLE DROP COLUMN
- Get the whole object: SELECT *

critique

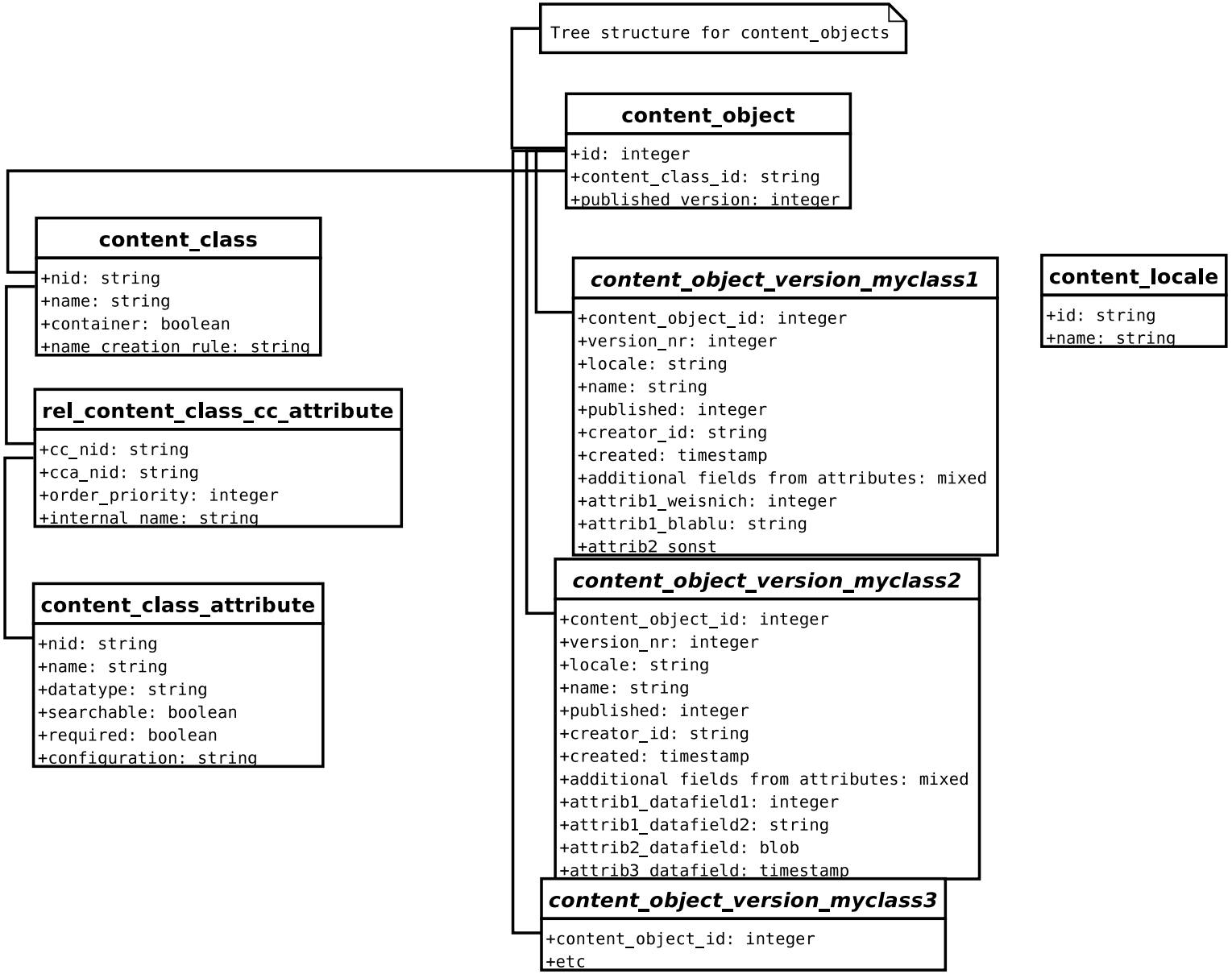
DDL is not transactional save

- MyISAM also does not have transactions and is used in production
- You do not change your content classes weekly but in a staged development process
- The current implementation in eZ Publish is even worse: PHP updates the related attributes until it times out

critique

You must access many tables to get informations on different object types

- Yes, but most of the time you instantiate the objects anyhow and can get the informations in PHP land
- Use views


Tree structure for content_objects
content_object
+id: integer
+content_class_id: string
+published_version: integer
content_class
+nid: string
+name: string
+container: boolean
+name_creation_rule: string
content_object_version_myclass1
+content_object_id: integer
+version_nr: integer
+locale: string
+name: string
+published: integer
+creator_id: string
+created: timestamp
+additional fields from attributes: mixed
+attrib1_weisnich: integer
+attrib1_blablu: string
+attrib2_sonst
content_locale
+id: string
+name: string
rel_content_class_cc_attribute
+cc_nid: string
+cca_nid: string
+order_priority: integer
+internal_name: string
content_object_version_myclass2
+content_object_id: integer
+version_nr: integer
+locale: string
+name: string
+published: integer
+creator_id: string
+created: timestamp
+additional fields from attributes: mixed
+attrib1_datafield1: integer
+attrib1_datafield2: string
+attrib2_datafield: blob
+attrib3_datafield: timestamp
content_class_attribute
+nid: string
+name: string
+datatype: string
+searchable: boolean
+required: boolean
+configuration: string
content_object_version_myclass3
+content_object_id: integer
+etc

conclusion

Databases are powerful tools, but todays PHP applications use only a small subset of the available features.

Outline

Linking is essential

. . . ERP5 also implements some extra features to enhance programming productivity. Perhaps the most interesting is the concept of relationship managers, which are objects responsible for keeping the relationships between pairs of objects.

[1]

[1] ERP5: Designing for Maximum Adaptability. In: Beautiful Code, S.345

Links in eGroupWare

```
'egw_links' => array(
 'fd' => array(
  'link_id'=>array('type'=>'auto','nullable'=>False),
  'link_app1'=>array('type'=>'varchar','precision'=>'25','nullable'=>False),
  'link_id1'=>array('type'=>'varchar','precision'=>'50','nullable'=>False),
  'link_app2'=>array('type'=>'varchar','precision'=>'25','nullable'=>False),
  'link_id2'=>array('type'=>'varchar','precision'=>'50','nullable'=>False),
  'link_remark'=>array('type'=>'varchar','precision'=>'100'),
  'link_lastmod'=>array('type'=>'int','precision'=>'8','nullable'=>False),
  'link_owner'=>array('type'=>'int','precision'=>'4','nullable'=>False)
```

Linking of contents must be a first class priority for the storage layer.

- Put 1:n links inside the table to reduce joins
- Individual n:m link tables for every pair of linked tables?

Outline

Surrogate Keys vs. Natural Keys

Surrogate Keys

‘id‘ int(11) NOT NULL auto_increment

Natural Keys

unique identifiers with meaning:

- en, de, ro, fr, . . .
- postal codes
- user logins
- email addresses

On surrogate keys

.. They suck.

- surrogate keys give no clue on the object they represent
- most of the times they are another unique index that needs to be checked by the DB
- are not portable from one system to another (dev - production)

Example

Content classes and class attributes in eZPublish have unique string identifiers and surrogate keys.

On natural keys

.. They are lovely.

- gives a hint for debugging and avoids wrong numbers
- often save an id column
- are portable from one system to another (dev - production)

Natural keys may include multiple columns.

on speed

- Don't overoptimize the DB! Do caching!
- Use binary instead of text: No colation
- int takes four bytes and all of them need them need to be compared, three letters can already be enough to identify a natural key!

links

- http://archives.postgresql.org/pgsql-hackers/2006-01/msg00414.php
- http://pooteeweet.org/blog/1015
- http://en.wikipedia.org/wiki/Surrogate_key