

# The FFII Database

Hartmut PILCH

<http://a2e.de/i2p/ffiidb>

April 17, 2009

The database scheme on which many social software applications at FFII are built. How it works. How it is being used.

## Contents

<b>1</b>	<b>Tables relevant for Member Managment</b>	<b>2</b>
1.1	Tables with key-set consisting of user only . . . . .	2
1.1.1	pnom: user name and primary uid definition . . . . .	3
1.1.2	pass: password . . . . .	3
1.1.3	mf: male or female . . . . .	3
1.2	Tables with key-set consisting of user and another distinctive feature . .	3
1.2.1	asoc: desired degree of involvement in FFII . . . . .	3
1.2.2	mail: e-mail addresses of the user . . . . .	4
1.2.3	padr: physical addresses of the user . . . . .	4
1.2.4	purl: web pages of the user . . . . .	4
1.2.5	teln: telephone/fax numbers of the user . . . . .	4
1.2.6	pag: payment terms . . . . .	4
1.2.7	konto: bank accounts . . . . .	4
1.2.8	firma: extra info from corporate members . . . . .	4
1.2.9	memq: membership in other organizations . . . . .	4
1.3	Tables with key-set consisting of elements other than the user . . . . .	5
1.3.1	asolack: user status acceptance event . . . . .	5
1.3.2	org: organizations related to the FFII or the FFII's work . . . . .	5
<b>2</b>	<b>Tables for Projects/Workgroups: proj, projhlp, projdon</b>	<b>5</b>
<b>3</b>	<b>Tables for peripheral applications</b>	<b>6</b>
3.1	Adopt a Repersentative: repr . . . . .	6
3.2	Participate in an Event . . . . .	6
3.3	Book a bed . . . . .	6

3.4	Adopt a Patent . . . . .	6
3.5	Watch a webpage . . . . .	6
3.6	Tables for campaigns . . . . .	6
3.6.1	aktbb: details about fundraising-contacts for an office in Berlin . .	6
<b>4</b>	<b>Meta-Tables</b>	<b>6</b>
4.1	Definition of allowable keys . . . . .	6
4.2	Scheme Description . . . . .	6
<b>5</b>	<b>Tables for (multilingual) document managment</b>	<b>6</b>
<b>6</b>	<b>Changes to be envisaged</b>	<b>6</b>
<b>7</b>	<b>What to do about supporters who want no more mail?</b>	<b>7</b>
7.1	Example: adding truelynx as active member, deprecating alias users . . .	7
<b>8</b>	<b>ActiveMembersGroup</b>	<b>8</b>
<b>9</b>	<b>Resources</b>	<b>9</b>
<b>10</b>	<b>History</b>	<b>9</b>
<b>11</b>	<b>Design Principles of the Database Schema</b>	<b>9</b>

Relational data, editable by the Standard Query Language (SQL), are likely to form a relatively stable informational “base” of a community. On this base, people collaborate and build applications. The core part of the FFII database applies a set of design principles to further facilitate application programming. Peripheral tablespaces such as the ledger tables of the former treasurer differ for historical reasons.

## Contents

### 1 Tables relevant for Member Managment

#### 1.1 Tables with key-set consisting of user only

In these tables the key-set consist only of the user ID uid, so that there is one value-set for each user.

##### 1.1.1 pnom: user name and primary uid definition

key-set

uid 8 letter acronym for each user, defined in this table

value-set

pnom personal name; if it contains a family name, that should be in capital letters if possible

pntyp type of personal name, one of several choosable types; references column ‘pntyp’ of table ‘pn

### 1.1.2 pass: password

key-set  
uid 8 letter acronym defined in table **pnom**  
value-set  
pass password, one-way-encrypted with crypt(3) function of the C Library

In the future stronger one-way encryption algorithms may be used for the password. The applications will have to try out different alternatives before raising an exceptions.

### 1.1.3 mf: male or female

key-set  
uid user ID; defined in table **pnom**  
value-set  
mf m (male) or f (female); references column 'mf' of table 'mf', in which its admissible values are

## 1.2 Tables with key-set consisting of user and another distinctive feature

### 1.2.1 asoc: desired degree of involvement in FFII

Not every user in **pnom** has an entry in **asoc**. There may exist several entries, one for each organization. Level of involvement may differ for one uid but different organizations.

key-set  
uid user ID; defined in table **pnom**  
org organization; default: 'ffii', may be used for special entries about relation to partner org  
value-set  
asolvl level of involvement; 3 active member, 2 passive member, 1 supporter, 0 user, -1 non-user  
asorem user's remark about his stated wish  
asolack latest acceptance event, refers to column **asolack** in table **asolack**  
editp boolean: whether we grant this user preliminary editing rights to ffii sites even without

### 1.2.2 mail: e-mail addresses of the user

key-set  
uid user ID; defined in **pnom**  
mailref type of mail address; default 'norm', other possible types, defined in 'mailref' field of 'mail'  
value-set  
mailuser the part on the left side of the '@'  
mailhost the part on the right side of the '@', for inactive addresses there is an '.inactive' postfix to

Mail addresses that should not be used any more, e.g. because the user does not want to receive any e-mails any more, should be invalidated by adding the postfix '.inactive' to the mailhost field. Some also have been invalidated with an '.invalid' postfix.

Note that this is not reliable, **asoc.asolvl** must also be checked before one sends mail to an address, also note that a user may have multiple mail addresses and one mail

address may be contained multiple times, e.g. because someone has registered himself multiple times.

### **1.2.3 padr: physical addresses of the user**

### **1.2.4 purl: web pages of the user**

### **1.2.5 teln: telephone/fax numbers of the user**

### **1.2.6 pag: payment terms**

whether and at what times and intervals we can expect or debit money from a member's account

### **1.2.7 konto: bank accounts**

### **1.2.8 firma: extra info from corporate members**

Economic Majority makes intensive use of this table.

### **1.2.9 memq: membership in other organizations**

May contain information about membership status in organizations that are in some way related to what the FFII does; this contains, e.g., membership status from the participation system (see Resources).

key-set

uid 8 letter acronym defined in table pnom

org 8 letter acronym defined in table org where org.memq is true

value-set

memlvl membership status of the individual in the organization: 0 - not a member, 1 - simple member

## **1.3 Tables with key-set consisting of elements other than the user**

### **1.3.1 asolack: user status acceptance event**

key-set

asolack acceptance event ID

value-set

uid user

ackdat date of acceptance

acklvl level to which his accession was accepted

ackuid the user who executed the decision on behalf of the organization

### 1.3.2 org: organizations related to the FFII or the FFII's work

key-set

org            8 letter acronym of organization

value-set

memq        wether this is an organization to be used in memq

eurolinus    wether this is organization was part of the eurolinux alliance

## 2 Tables for Projects/Workgroups: proj, projhlp, projdon

- proj defines projects/workgroups, see d++ proj
- projhlp lets users state their wish to join a project at a certain level, analogous to asoc, see d++ projhlp
- projprm lets project owners limit the level at which they permit participation of individual users, similar to asolack but not event-based. If a user is accepted at hlpvl  $\geq 3$  and prmlvl  $\geq 3$ , then he should be automatically subscribed to the -help, -parl and -news mailing lists. see d++ projprm
- projdon, used for fundraising of projects/workgroups

This system is quite sufficient for basic workgroup operation and is used in the aktiv system for . Unfortunately much of it's functionality, such as the mechanism for subscription to mailing lists, was been disabled by the FFII sysadmin in 2006.

## 3 Tables for peripheral applications

### 3.1 Adopt a Repersentative: repr

### 3.2 Participate in an Event

### 3.3 Book a bed

### 3.4 Adopt a Patent

### 3.5 Watch a webpage

### 3.6 Tables for campaigns

#### 3.6.1 aktbb: details about fundraising-contacts for an office in Berlin

A table created for a campaign to raise funds for an office in Berlin, contains only those that have been contacted, time and result (if any) of the contact.

## 4 Meta-Tables

### 4.1 Definition of allowable keys

### 4.2 Scheme Description

## 5 Tables for (multilingual) document management

## 6 Changes to be envisaged

Editing (update/insert) the database is easier when tables are unified. Tables that have the same keys, such as 'pnom', 'pass' and 'mf', might as well be merged. There are also some advantages (as well as disadvantages) to merging the primary e-mail address, physical address, homepage as well as the 'asoc' information into this single table.

There is also a discussion as to whether we want more descriptive naming for tables and columns. This could make application programs easier to read and maintain. The downside of descriptive namings is that

- really descriptive names would have to be very long
- meta tables and applications become bigger
- more typing needed, more opportunities for misspelling
- full understanding can only be gained from documentation strings (invocable with `\\dd+` and `\\d+`); when people rely on descriptive naming, half-understanding becomes the norm

Changes to the database scheme should be planned and carried out in a similar way as changes to the statute: carefully and with sufficiently long transition periods.

Any member management system should be modular enough to make such changes easy to carry out once they are decided.

This can be achieved by encapsulating all SQL calls into functions which adequately describe what the application does, and collecting these function definitions in a single interface library file.

## 7 What to do about supporters who want no more mail?

There will usually still be a reason to keep minimal data, especially those that relate to the identity of the person.

E.g. we will need that when asked questions about history, e.g. how many members we had when, who signed a certain petition etc.

The solution is fairly simple: those who do not want mail from us are not fffi supporters. Their asolvl should be 0, and we can further clarify the decision from our side by adding an asolack record with acklvl = 0.

We never send out mails to anyone who is not a supporter.

If we doubt our own ability to adhere by our rules, we can of course also consider appending an '.invalid' to the mail.mailhost, but that would impair the functionality of any member management system that relies on mail addresses for certain identity-related functions (such as change password, change user id, change mail address – those require confirmation mail).

## 7.1 Example: adding truelynx as active member, deprecating alias users

The following is based on a mail sent by phm to polis-help on 2008-01-19.

Meanwhile also truelynx is registered as an active member, see the SQL dialog below.

```
ffii=# insert into asoc ( uid, asolvl ) values (
'truelynx', 3 )
INSERT 0 1
ffii=# insert into asoc ( uid, asolvl ) values (
'rpfeiffx', -1 )
INSERT 0 1
ffii=# begin; update asoc set asolvl = -1, asorem
= 'alias for truelynx' where uid in ( 'pfeiffep',
'rpfeiffx' )
BEGIN
UPDATE 2
ffii=# commit
COMMIT
ffii=# insert into asolack ( asolack, uid, ackdat,
acklvl, ackuid, ackrem ) values (
'truelynx080119', 'truelynx', '2007-12-01', 3,
'phm', 'truelynx became active member on the day
of the GA 2007 in Brussels and was elected reserve
board member there. He needs to be in
ActiveMembersGroup and TrustedGroup ASAP' )
INSERT 0 1
ffii=# begin; update asoc set asolack =
'truelynx080119' where uid = 'truelynx'
BEGIN
UPDATE 1
ffii=# commit
COMMIT
```

I guess you understand what I did there:

1. degraded the pfeiffep and rpfeiffx records to asoc.asolvl -1. However they can't be deleted yet for reasons of historical consistency (e.g. signed calls for action) and because some of the contact data are richer there than under the truelynx identity
2. added an asoc record for truelynx, pretending that truelynx himself wants to be an active member
3. added an asolack (member status acceptance event) record: accepting truelynx in the name of FFII.
4. added a reference to the asolack event into asoc.

If the secretary were to disagree, he'd do steps 3 and 4 again without deleting my asolack record.

The last step (adding the asolack reference to asoc) could be done automatically by a cron-driven script which regularly updates asoc so that the asolack reference is the newest one, i.e. the one with maximal asolack.ackdat.

## 8 ActiveMembersGroup

The list of people who are authorised to read/write internal ffii wiki pages is computed as follows:

```
ffii=# select count(*) from asoc natural join
asolack where editp or (asolvl >= 3 and acklvl >=
3)
```

## 9 Resources

- Identifying FFII users for proper bookkeeping – some hints on how to deal with typical situations of user management that occur during treasury work
- MLHT Database – some old documentation
- inline documentation in the PostgreSQL database itself
- <http://old.ffii.org/assoc/verbdefs.txt>: database scheme and migration steps (mov) described in a meta-language used by the Knecht.pm library
- aktiv: the current participation system
- PHP Users: adapted version of `php_users` in which all SQL commands are encapsulated in function calls that describe the application logic rather than the database logic. One more work day should get this working with the FFII database.



## **10 History**

- 2007-03-12 phm creates this directory.

## **11 Design Principles of the Database Schema**

Moved. See FFII Database Design Principles.