# Agenda

- Introduction of design and work flow objectives
- Overview support classifications
- Architecture of database
- Workflow examples & demonstration
  - Introduction of new member
  - Entry of funding receipts
  - Initiation of membership renewals
- Database Conversion
- Implementation Discussion

# Design Objectives

- Use existing concepts from the Access based system to retain familiarity.
  - Concept of the Member/Contact IDentifier (MCID)
  - Use existing database constructs of
    - Member/Volunteer
    - Funding
    - Correspondence
    - Time Entry (for volunteers)
- Integration of members, volunteers, volunteer mail utility and (eventually) bulk mail utility to use single database.
- Introduce data integrity checks to enforce data entry integrity.
- Provide flexible tools for management and reporting.
- Utilize GPL (free) software for all system components.
    - MySQL – database
    - PHP – programing language
    - HTML5 -web page construction
    - Javascript – page level field validation
    - Bootstrap – page layout
    - phpMyAdmin – database administration
    - LIbreOffice

# New Functions Available

All with common user interface
All available to access via multiple platforms
All integrated with the same database
Alll implemented with GPL software

- Membership Database
- Volunteer Database
- Generic Database Lookup

# Support Community Definitions

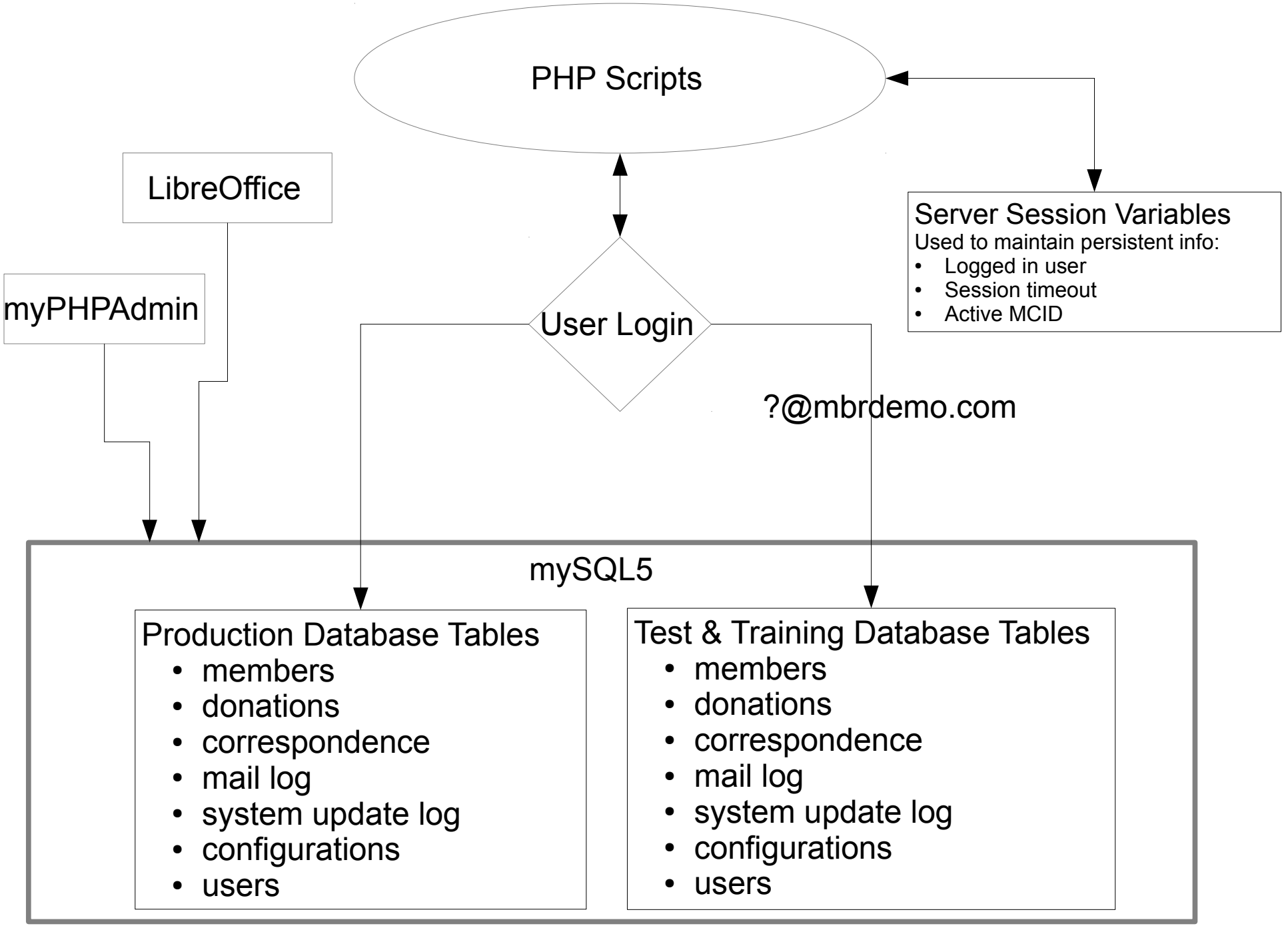
- Similar (but different) that used in Access system.
- Classifies supporter 'groups' or 'classes'

**Members** - supporters that provide scheduled funding payments.

**Volunteers** – supporters that provide scheduled funding payements AND contribute volunteer time in some capacity.

**Donors** – provide sporadic funding support and are usually organizations, groups, trusts.

**Contacts** – none of the above. This is the 'pool' of interested potential supporters available to be recruited into one of the other categories.


PHP Scripts
LibreOffice
myPHPAdmin
Server Session Variables
Used to maintain persistent info:
• Logged in user
• Session timeout
• Active MCID
User Login
?@mbrdemo.com
mySQL5
Production Database Tables
• members
• donations
• correspondence
• mail log
• system update log
• configurations
• users
Test & Training Database Tables
• members
• donations
• correspondence
• mail log
• system update log
• configurations
• users

# User Levels – Membership Database

- User – all normal user functionality
  - Add new member
  - Maintain member information
  - Entry of new funding and correspondence records
  - Create reminders
- EDI user – User level plus Extended Donor Info menu items.
- Admin user – EDI user plus extended system administrative functions menu.

# User Levels – Volunteer Database

- User – all regular user functions including:
  - Entry of a new system user
  - Entry of a new volunteer
  - Update volunteer information
  - Maintenance of volunteer mailing lists
- Admin user – User plus
  - Sending email to list(s) and
  - Creation of new list

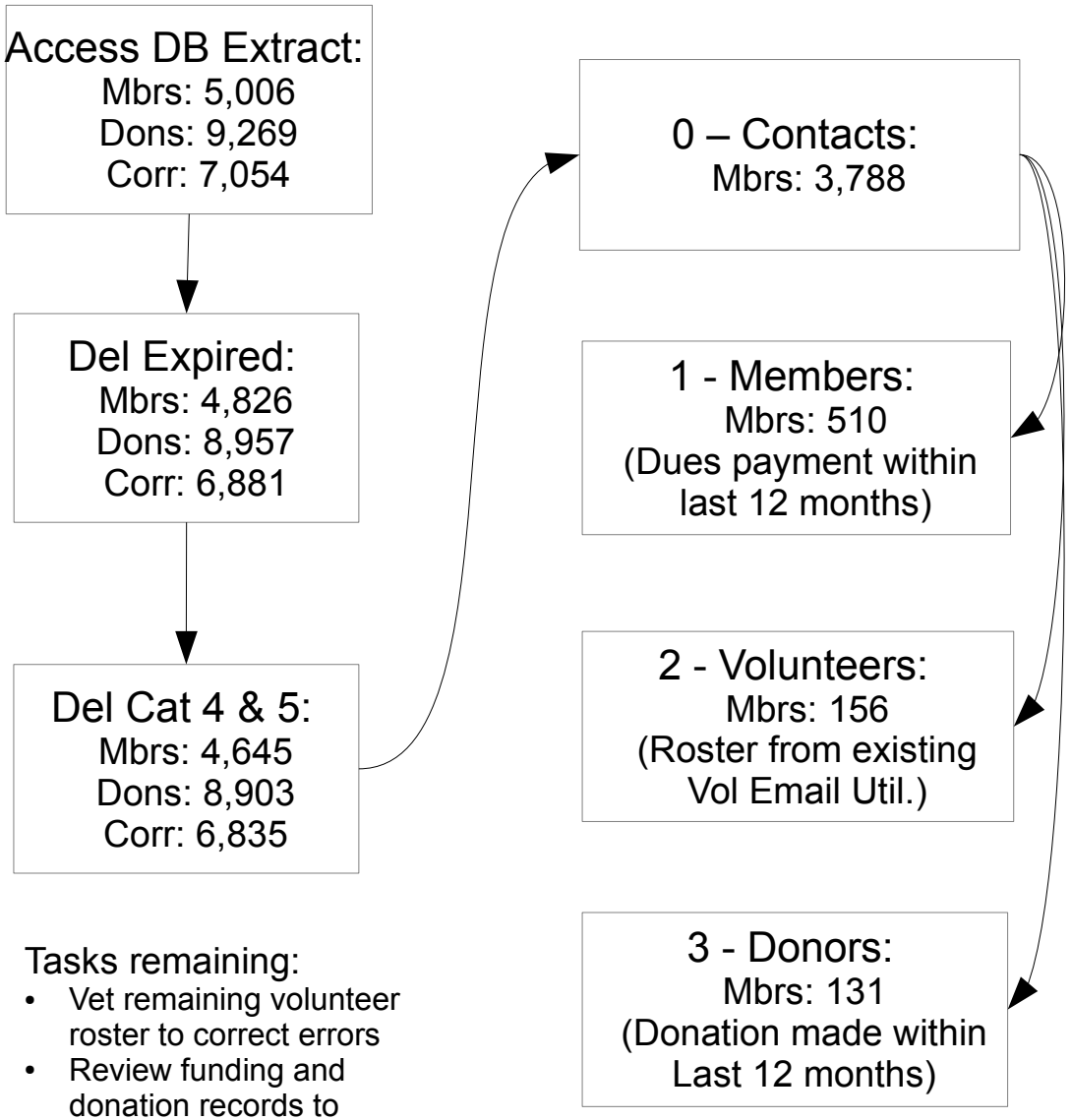
# Workflow examples & demonstration

Introduction of new member

Funding & Correspondece Entries
(Dues, Donations, Letters, Postcards, Other)

Initiation of membership renewal(s)
(Identify, email/mail notices, print labels/letters,reports)

# Database Conversion




Tasks remaining:

- Vet remaining volunteer roster to correct errors
- Review funding and donation records to determine classifications

## Methodology

1. Extract all member, donation and correspondence records from Access database into a series of spreadsheets.
2. Format spreadsheets adjusting column names and values.
3. Set status of all records = 0.
4. Import spreadsheets into database.
5. Run database utilities to evaluate funding records to determine status values of 1 and 3.
6. Run database utility to merge vol email roster to determine status value of 2.

## Admin Functions

- Administer users
- Maintain configuration tables (drop down definitions)
  - Member types
  - Funding purposes
  - Funding programs
  - Funding campaigns
  - Correspondence types
  - Volunteer email lists
  - Volunteer categories
- Delete records (only when absolutely necessary)
- Administer templates (usually done by membership chair)
- Do DBJanitor page

## Database Backup Strategies

- ISP does server back-ups, BUT
  - Export entire production db to spreadsheet(s) or sql scripts?
  - Copy of entire db to server back-up copy?

## Script Update Strategy

- Rename current server directory to 'dirname.old'
- Copy entire new server directory from development environment
- Updates everything including all documentation to on-line server