Disclaimer:

I usually present this software to non VO-savvy people
What is VODb?

A desktop application that aims to simplify the process of querying astronomical databases
A little history

- Started life as part of the Pan-STARRS project
- Pan-STARRS has a lot of data* and a complex schema
- Difficult for the novice to retrieve any meaningful data
- (Difficult for experts to retrieve any meaningful data)
- Many queries would never complete

* time-domain; 150 billion detections; 5.5 billion sources; >10xSDSS in first year
Existing web interface

- Pan-STARRS has an SDSS-like web interface
- Offers a query box and a basic query builder

but...

- Web-pages offer limited functionality
- Results need to be manually downloaded...
- ...then loaded into a desktop application for analysis
Objectives of VODb

- To make it easier to generate SQL
- To utilise existing VO software to analyse the results
- To provide the ability to work on data offline

In short: to *enable* science
Aren't there existing VO database access tools?
Yes, but...

Tools like (the excellent) TOPCAT enable database querying as an *extra*. VODb is a *dedicated* astronomical database access tool that tries to apply the UNIX philosophy of:

"*Do one thing and do it well*"
Nearly ideal, but...

- Can't connect to Pan-STARRS (or other non-TAP Dbs)
- Doesn't provide the higher level abstraction we wanted
- We wanted a more 'fancy' drag 'n drop query builder
What's so difficult about SQL?
Lots of things are difficult about SQL

- SQL is a programming language (an ancient one)
- Not all astronomers are programmers
- Standardised, but too late: dozens of dialects
- Most vendors don't support the entire standard
- Most vendors diverge from the standard (e.g., date/time)
- Vendors have an incentive to remain non-standard
But it's *declarative*, which is good...

● In theory we should just ask for *what* we want
● We should not have to specify *how* to get it

*but...*

● Bad performance often enforces an imperative approach
● Requires users to have knowledge of the *implementation*

This is definitely *bad*
But we're stuck with SQL
so...
Whenever possible, create an abstraction layer so that knowledge of the schema, indexes, keys, partitioning etc is not necessary

When it is necessary to write SQL directly, provide as much help as possible (user-friendly GUIs)
What data can you access?

- Any local JDBC-supported database
- Any remote TAP database (SDSS, 2MASS etc)
- Pan-STARRS (registered users only)
- A local *Derby* database to store results
- Design allows new connection types to be added easily
VO compliant

- SAMP
- VOTable
- Table Access Protocol (TAP)
Some features
A user-friendly double-click desktop application
A straightforward, searchable connection manager

<table>
<thead>
<tr>
<th>Connection</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESMunich</td>
<td>The DES database in Munich</td>
</tr>
<tr>
<td>Derby</td>
<td>Your local Derby database</td>
</tr>
<tr>
<td>FIRST</td>
<td>FIRST Survey Catalogue (03Apr11 Version)</td>
</tr>
<tr>
<td>GLIMPSE</td>
<td>Galactic Legacy Infrared Mid–Plane Survey Extraordinaire</td>
</tr>
<tr>
<td>IRAS</td>
<td>Infrared Astronomical Satellite Archive</td>
</tr>
<tr>
<td>Millennium</td>
<td>Millennium Simulation Database</td>
</tr>
<tr>
<td>Pan–STARRS</td>
<td>The Pan–STARRS database</td>
</tr>
</tbody>
</table>
SQL syntax highlighting and other text editing conveniences
Drag 'n drop graphical query builder
Astronomical date/time and coordinate converters
Mask generator e.g. for SDSS data quality flags
SAMP communication (pass data to TOPCAT etc)
Local Derby database including full query history
Query diagnostics to help users understand why they waited so long
But those features still require writing SQL
What are plug-in queries?

- Think of them as sophisticated sample queries
- (SDSS and PS: the “20 queries” design methodology)
- Curated online (in XML) for instant deployment
- Available through menus
- Delivered as user-friendly GUIs
- Constrained parameters reduce errors
- Default values enable quick, successful queries
Plug-in queries
Why plug-in queries?

- Using a query builder requires a thorough knowledge of the schema
- Sample queries are helpful, but cumbersome (copy-and-paste, edit)
Example plug-in query XML

<query>
  <author>Millenium</author>
  <title>Tully-Fisher relation</title>
  <shortDescription>Find the Tully-Fisher relation...</shortDescription>
  <longDescription>Find the Tully-Fisher relation, Mag vs Vvir, for galaxies with a bulge/total mass ratio less than a given value</longDescription>
  <param>
    <name>PARAM_BULGE_MASS_RATIO</name>
    <description>Bulge/total mass ratio</description>
    <type>float</type>
    <max>1</max>
    <min>0</min>
    <default>0.1</default>
  </param>
  <param>
    <name>PARAM_SNAPNUM</name>
    <description>Snapnum (redshift)</description>
    <type>int</type>
    <max>63</max>
    <min>0</min>
    <default>41</default>
  </param>
  <sql>SELECT vVir, mag_b, mag_v, mag_i, mag_r, mag_k
         FROM millimil.DeLucia2006a
         WHERE (bulgeMass < PARAM_BULGE_MASS_RATIO*stellarMass OR bulgeMass IS NULL)
         AND snapnum = PARAM_SNAPNUM</sql>
</query>
Learning by example

- It's difficult to write good SQL
- Plug-ins allow quick, successful querying
- Allow users to adapt, enhance and learn by example
- Help users improve SQL skills
- Help users and become familiar with schema
VODb key principle: Abstraction

Astronomers shouldn't have to worry about:

- Database implementation (indexing, partitioning etc)
- File formats (FITS, csv, VOTable etc)
- Connection types (TAP, JDBC, Casjobs etc)

These are distractions from the science...
VODb key principle: Abstraction

Instead:

- Users given a list of databases with meaningful names
- Users do not interact with any files
- Reduced need to understand the peculiarities of SQL
Some users

- Popular with Pan-STARRS users
- Some interest from LSST and VAO
- Interest from Millennium simulation database
- Possibility to use with Euclid prototype archive
http://goo.gl/4X147