Delicious Data:
Automated Techniques for Complex Reports

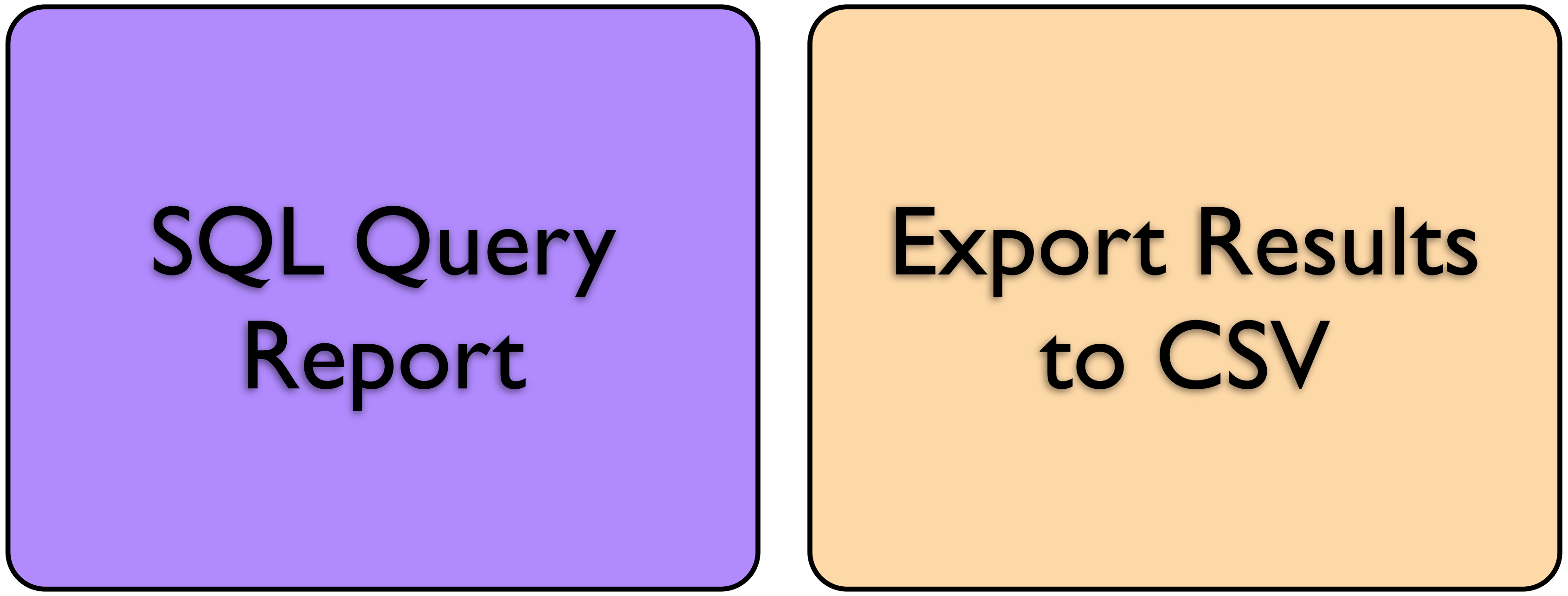
Presented by
Jeff Godin and Benjamin Shum

How NOT to run reports...

Planning the framework

Planning the framework

Planning the framework



Planning the framework

SQL Query Report

Export Results to CSV

Convert CSV to XLS

Planning the framework

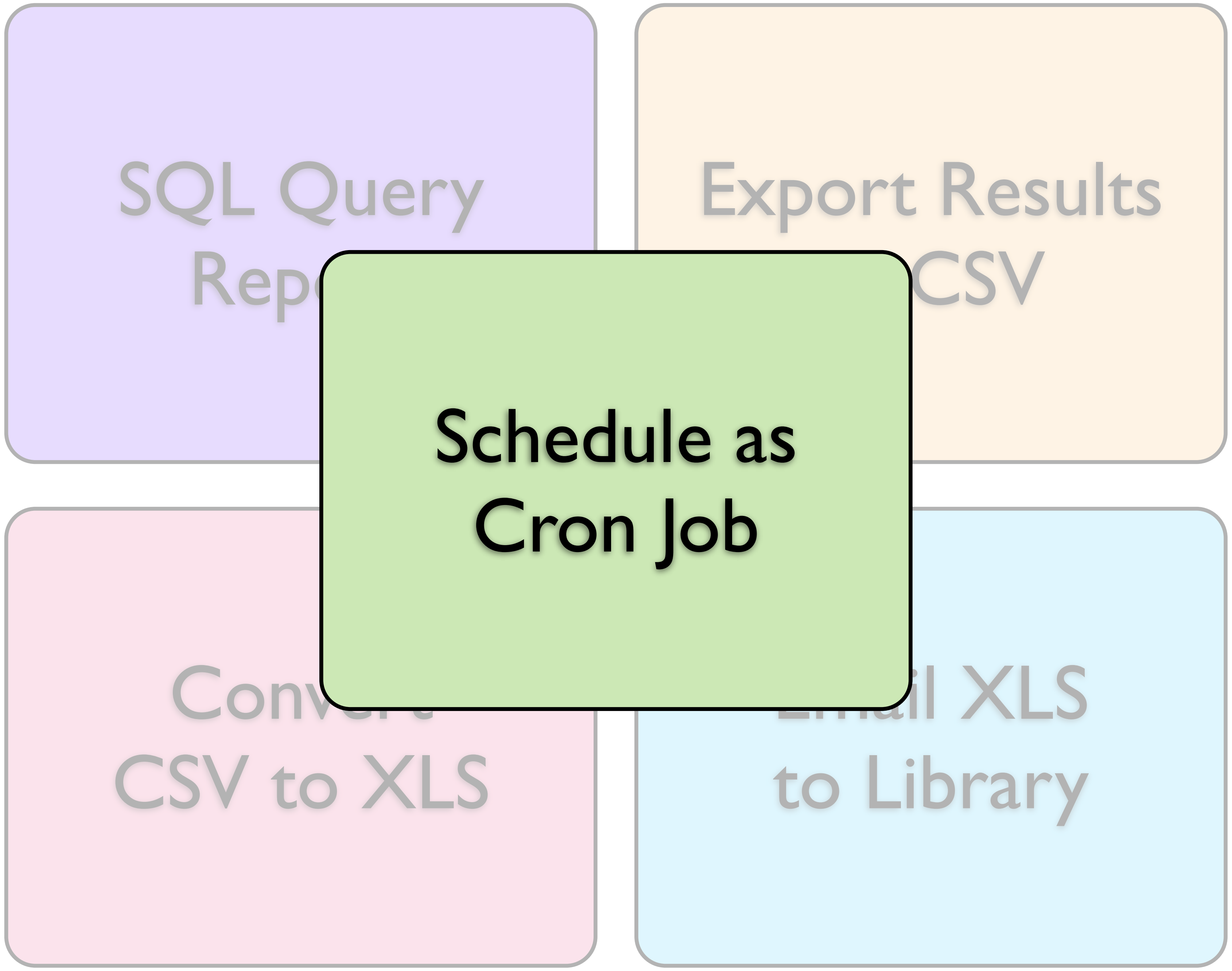
SQL Query Report

Export Results to CSV

Convert CSV to XLS

Email XLS to Library

Planning the framework



Some Strategies

Schedule as Cron Job

Use our experience setting up crontab

SQL Query Report

Use our existing SQL reports

Export Results to CSV

CSV is a standard format to work with

Convert
CSV to XLS

Find a converter

Email XLS
to Library

Find an email client

Implementation

- Can find online at http://evergreen-ils.org/dokuwiki/doku.php?id=scratchpad:automated_sql_reports

Future of the work

- Additional formatting
- Dealing with reports with zero results
- Adding new reports
- Implementing more refinements

No database access?

- Create scheduled reports
- Completion notification via e-mail
- Script pulls report data
- Import and process

Don’t do this unless you have to

Reports as Exports

- Create template that outputs raw data
- Schedule report
- Enable e-mail notification to special address

Act on the e-mail

- fetchmail -> procmail -> perl
- Report data is automatically downloaded
- Within a minute or so

act-on-mail.pl

```
#!/usr/bin/perl

# called by procmail, and receives the mail message on stdin

use strict; use warnings;

my $url;

while (<>) {
        if ($_ =~ /https:\/\/reporter/) {
                $url = $_;
                last;
        }
}

system("/home/stats/bin/fetch-report.pl", "-u", $url);
```

fetch-report.pl logic

- Fetch URL like:
  https://host/reporter/[...]/report-data.html
- Automatically log in if needed (keep a cookie)
- Download report-data.csv and save to output dir

imports.ini example 1

```
[general]
dir=/home/stats/reports
state_dir=/home/stats/state

[group holdratio]
name=Hold Ratio
file1=holds_outstanding.csv
file2=holdable_copies.csv
file3=bib_info.csv
command=psql -f /home/stats/sql/holds_init.sql
#post-command=/home/stats/bin/mail-holdratio.pl
```

imports.ini example 2

```
[general]
dir=/home/stats/reports
state_dir=/home/stats/state

[group items_for_export]
name=Items for Export
file1=items_for_export.csv
command=psql -f /home/stats/sql/items_for_export.sql
post-command=/home/stats/bin/export_bibs.pl -f /home/stats/
output/bre_ids.csv
```

Import script overview

- Runs from cron
- Have all input files been updated since the last time I ran this import?
- Run import
- Update timestamp
- Run post-import command

The import process

- drop table
- create table
- import data
- process to output

The post-import

- Send the report output somewhere
- E-mail notify that "report's ready at someurl"
- Trigger something like a bib export of the records that were output

One more thing...

- Don’t forget to delete your scheduled report output

Custom reporting views exposed in the IDL

SQL views in the IDL

- Exposed via reporting interface
- Good for on-demand reporting
- Don’t break the IDL
  - xmllint is your friend
  - pick a class id unlikely to collide

```
<class id="rlcd"
controller="open-ils.cstore open-ils.pcrud open-ils.reporter-store"
oils_obj:fieldmapper="reporter::last_copy_deleted"
oils_persist:readonly="true"
reporter:core="true"
reporter:label="Last Copy Delete Time" >
```

```
<oils_persist:source_definition>

SELECT  b.id,
        MAX(dcp.edit_date) AS last_delete_date

FROM   biblio.record_entry b
        JOIN asset.call_number cn ON (cn.record = b.id)
        JOIN asset.copy dcp ON (cn.id = dcp.call_number)

 WHERE  NOT b.deleted

 GROUP BY b.id

 HAVING SUM( CASE WHEN NOT dcp.deleted THEN 1
ELSE 0 END) = 0

</oils_persist:source_definition>
```

```
<!-- continued -->
      <fields oils_persist:primary="id"
oils_persist:sequence="biblio.record_entry">
         <field reporter:label="Record ID" name="id"
reporter:datatype="id"/>
         <field reporter:label="Delete Date/Time"
name="last_delete_date" reporter:datatype="timestamp"/>
      </fields>
      <links>
         <link field="id" reltype="has_a" key="id" map=""
class="bre"/>
      </links>
```

```
<!-- continued -->
        <permacrud xmlns="http://open-ils.org/spec/opensrf/
IDL/permacrud/v1">
            <actions>
                <retrieve/>
            </actions>
        </permacrud>
    </class>
```

The Future

- General third party reporting frameworks?
- Library Dashboards?
- Your Ideas?

Photo Credits

Surrounded by papers by Flickr user suratlozowick

http://www.flickr.com/photos/suratlozowick/4522926028/

Thanks!