

# PSUG National Information Exchange

## Users Helping Users



The logo for Flamingo Las Vegas is located in a red rounded rectangle. The word 'Flamingo' is written in a large, white, cursive script font, and 'LAS VEGAS' is written in a smaller, white, all-caps sans-serif font below it.

# PowerQueries

## And the Data Export Manager

By Jim Parsons and Roger Sprik

[jparsons@vcschools.org](mailto:jparsons@vcschools.org)

[rsprik@vcschools.org](mailto:rsprik@vcschools.org)



# Test Server Available

<https://ps9.psugcal.org/admin>

Admin User: #aghs1 ... where # is a number from 1 to 99

Password: aghs1 ... same password for all users

# Using PowerQueries

## Why use PowerQueries?

1. Data Export Manager - custom exports (including scheduled outputs)
2. Internal API - via AJAX
3. External API - from a 3rd Party Server (OAuth)
  - a. Requires OAuth and Access Requests (not covered in this presentation)

## Demonstrations:

1. Creating a PowerQuery
2. Accessing via Data Export Manager

# Using `tlist_sql` to Get JSON data

```
[
~[tlist_sql;
  SELECT
    lastfirst,
    floor(months_between(sysdate, dob) / 12),
    to_char(dob, 'Month D, YYYY')
  FROM
    students
  WHERE enroll_status = 0
    AND floor(months_between(sysdate, dob) / 12) <= ~(gpv.max)
    AND floor(months_between(sysdate, dob) / 12) >= ~(gpv.min)
]
{
  "lastfirst": "~(name;json)",
  "age": "~(age;json)",
  "birthdate": "~(dob;json)"
},
[/tlist_sql]{}]
```

# Create a PowerQuery

1. PowerQueries are loaded via Plugins
2. Create three XML files:
  - a. ./plugin.xml
  - b. ./queries\_root/**example**.named\_queries.xml
  - c. ./permissions\_root/**example**.permission\_mappings.xml
3. ZIP the files, and install the Plugin.

See "PowerQuery Example 1.0" plugin

# The “plugin.xml” File

```
<?xml version="1.0" encoding="UTF-8"?>

<plugin xmlns="http://plugin.powerschool.pearson.com"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        description="Powerquery Example"
        name="PowerQuery Example (Birthdays)"
        version="1.0"
        xsi:schemaLocation="http://plugin.powerschool.pearson.com
plugin.xsd">

    <publisher name="Jim Parsons">
        <contact email="jparsons@vcschools.org"/>
    </publisher>

</plugin>
```

# The “example.named\_queries.xml” File

```
<queries>
  <query name="org.psugcal.example.students.birthdays"
        coreTable="students" flattened="false">
    <description>Birthday info for all Active Students -
        District Wide</description>
    <columns>
      <column column="students.student_number">id</column>
      <column column="students.lastfirst">name</column>
      <column column="students.grade_level">grade_level</column>
      <column column="students.dob">birthdate</column>
      <column column="students.dob">age_years</column>
      <column column="students.dob">month_day</column>
    </columns>
    <sql> <![CDATA[
        SQL query goes here
      ]]>
    </sql>
  </query>
</queries>
```



# PowerQuery SQL Statement

1. Only allows SELECT (or WITH . . . SELECT); read only
2. Columns with calculations MUST be aliased

```
<sql>
```

```
  <![CDATA[
```

```
    SELECT
```

```
      student_number,
```

```
      lastfirst,
```

```
      grade_level,
```

```
      TO_CHAR(dob, 'MM/DD/YYYY') AS birthdate,
```

```
      floor(months_between(sysdate, dob) /12) AS age_years,
```

```
      TO_CHAR(dob, 'mm/dd') AS month_day
```

```
    FROM
```

```
      students
```

```
    WHERE
```

```
      enroll_status = 0
```

```
    ORDER BY lastfirst
```

```
  ]]>
```

```
</sql>
```

# PowerQuery Columns

```
<columns>  
  <column column="students.student_number">id</column>  
  <column column="students.lastfirst">name</column>  
  <column column="students.grade_level">grade_level</column>  
  <column column="students.dob">birthdate</column>  
  <column column="students.dob">age_years</column>  
  <column column="students.dob">month_day</column>  
</columns>
```

SQL returns 6 columns, you must declare 6 columns

**SELECT**

```
  student_number,  
  lastfirst,  
  grade_level,  
  TO_CHAR(dob, 'MM/DD/YYYY') AS birthdate,  
  floor(months_between(sysdate, dob) /12) AS age_years,  
  TO_CHAR(dob, 'mm/dd') AS month_day
```

**FROM ...**

# The “example.permission\_mappings.xml” File

*(Only required for internal use)*

```
<permission_mappings>
  <permission name='/admin/home.html'>
    <implies allow="post">
      /ws/schema/query/org.psugcal.example.students.birthdays
    </implies>
  </permission>
</permission_mappings>
```

Basically, this says, “Whoever has access to “/admin/home.html” also has access to this PowerQuery. This applies only to internal API access via cookies, and DRF will constrain the results (and/or exclude records outside the scope of the user’s permissions or restrictions).

# Data Export Manager

PowerQueries are available in DEM

1. Exported data is constrained by the "Data Restriction Framework". <https://support.powerschool.com/article/74748>

DRF allows more students than what you're used to:

- students.schoolid
- cc.schoolid, reenrollments.schoolid, students.next\_school


2. The PowerQuery must not contain "arguments"
3. Columns with calculated data can NOT be filtered in DEM

# Data Export Manager

1. Scheduling saved templates is limited to users with a Role that allows it.
  - Create a Role or check the box to allow Scheduled Exports in an existing role.
  - Assign that role to the user.

## Edit User Access Role

Definition Export/Edit

 These access levels only apply to exporting, previewing, and editing with the Data Export Manager.

**Settings**

Default Access Level	<input checked="" type="checkbox"/> Export <input checked="" type="checkbox"/> Edit
User can set up and run scheduled system templates	<input checked="" type="checkbox"/>

2. Add Remote Connections (only SFTP supported)
  - Managed via "Remote Connection Manager" in Plugin Management

# Data Export Manager Security

Depending on the user's permissions, the Data Export Manager will constrain the results to the scope of the current school and the user's security restrictions, such as Field Level Security.

Once a template is scheduled for export, it becomes a system template and is available for exporting data associated only within the school context it is created. System templates created at the district office will export data from all schools.

Data exported from System Templates do not have field level security restrictions. However, users who are permitted to set up and run scheduled system templates do not necessarily have access to the export destination. Security of the export destination is managed by its own security mechanisms, outside of PowerSchool.

"Saved Templates" are user-based

"Scheduled Templates" can be accessed by any user with the proper role.

# Valley Christian PowerQueries

- JAMF "Classes" (Classroom App)
- Apple School Manager (Managed AppleIDs and Classroom App)
- HMH ThinkCentral (Lower Elementary Math Content)

These are custom-written for our use. They cannot be "dropped in" on your server, because they reflect our own decisions on how to implement the integration, custom fields you don't have, "hard coded" elements like `schoolid in (2,3)` or the use of regular fields that differ in how your district uses them.

They are provided as "real-life" examples of more complex queries.

**YOUR MILEAGE MAY VARY!!!!**

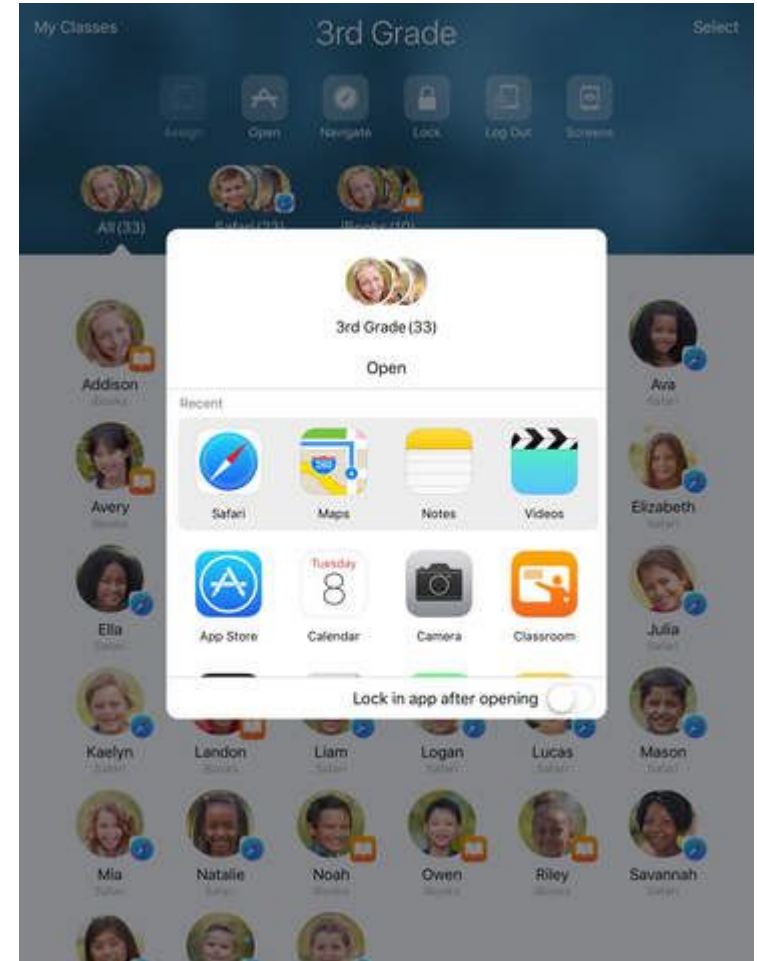
## JAMF -> "Mobile Device Management" software.

"jss\_classes\_1\_a.zip"

- Old "Focus" app
- Now supports Apple's "Classroom" app
- Our scheduled PowerQuery uploads the class enrollments every morning, because they can vary based on the bell schedule for that day.

With Classroom, teachers can:

- see what students are doing on their iPad
- lock the iPads to a particular app
- show the student iPad on the room's screen via AirPlay
- and more...

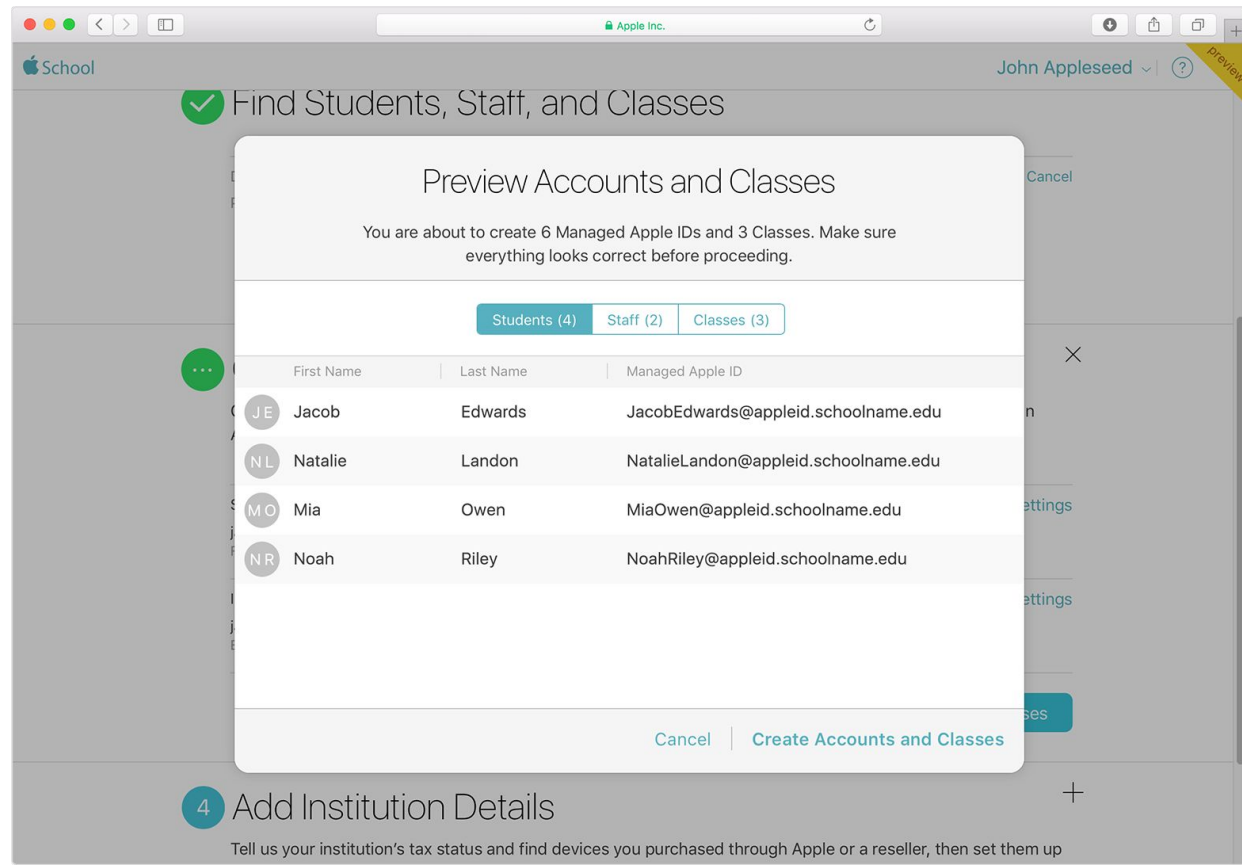




# Apple School Manager

"ASM\_1\_0.zip"

- Creates "Managed AppleIDs" for our 7th and 8th graders.
- Creates 6 files on the server every morning. A script then zips them up and uploads to an Apple SFTP server.
- We use WinSCP to script the SFTP upload and IZArc to script the creation of the zip file



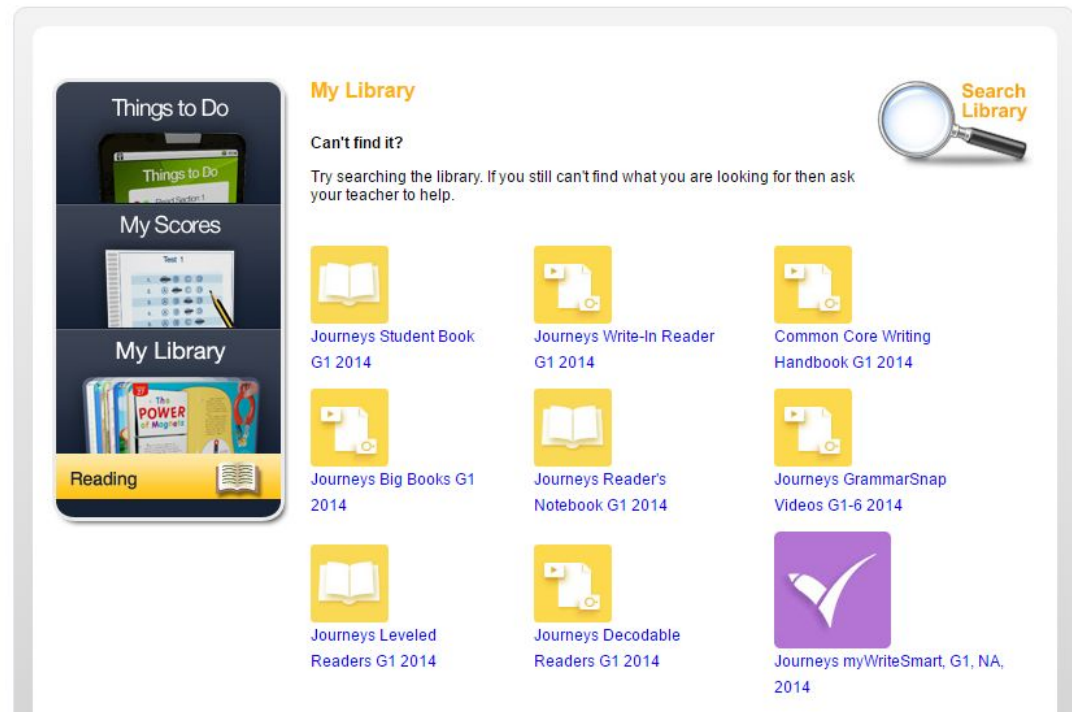
With managed AppleIDs:

- the school controls the ID
- the school manages app distribution
- available for students under 13

# HMH ThinkCentral

"hmh\_1\_0.zip"

- Creates classes, and enrollments in HMH ThinkCentral.
- Still use AutoSend for the student users because of the password issue
- Creates 3 files on the server every morning. A script then zips them up and uploads to an HMH SFTP server.



HMH ThinkCentral is an online portal. We're using it for their "Journeys" reading program.

# Questions?

Thank you for attending.

Resources:

<https://support.powerschool.com/developer> - a must for the PowerSchool customizer

<https://support.powerschool.com/article/74748> - The "Data Restriction Framework"

Jim Parsons (jparsons@vcschools.org)

Roger Sprik (rsprik@vcschools.org)