PowerSchool Enterprise Reporting

Presented by Roger Sprik - October, 2016 (as of PowerSchool 10.0.3)

Overview

In version 9.2, PowerSchool introduced a new reporting tool named “Enterprise Reporting”. It is an implementation of the reporting features already available in a platform from Oracle called “Oracle Application Express”, also known as “APEX”.

APEX is a mature product, with extensive documentation from Oracle, forums, tutorials and more. APEX is a platform for developing a complete application, but for PowerSchool we are focused on only a subset of APEX called “Interactive Reports”. Keep this in mind as you read the APEX documentation.

PowerSchool is referring to this as a “Preview Release”. Even with PowerSchool version 10, this means it needs more work, lacks complete documentation, and may change in future versions.

That being said, Enterprise Reporting is very powerful, very flexible, and customizing the reports is very visual, which results in greater ease of use. Some key features include:

- Flashback. See data in the past, even re-import it.
- Subscription. Have reports sent automatically on a schedule to email.
    - NOTE: As of 10.0.03 Email subscription is not supported and may not work.
- Flexibility. You can add your own reports or modify the included ones.

Answer Questions

Enterprise Reporting is for answering questions. Here are some real life examples of questions we’ve been able to answer with APEX in the short time we’ve been previewing it at my school.

- What are the average grades our Math teachers give, compared by course name?
- Which current seniors have taken the PSAT in the past?
- What are the PSAT 8/9 scores for our 8th graders, including the class averages, and the breakdown by test section of how many are on track for college readiness?
- What rooms are free in a given period?
- Where is each student in 2nd period today?
- Which students earned Ds or Fs in the last reporting period?

New in version 10

- Enterprise Reporting is now a tab on System Reports
- There is a new Enterprise Reporting Manager Role that gives more access to setup the Enterprise Reporting page
- Published/Unpublished status of reports

- Enterprise Reporting User Access Role tab for user, setup by categories
- Access to build APEX reports is an individual user permission, in their security settings.
- The username, name, and email are brought across to the APEX builder system, you must be logged into both PowerSchool and APEX

Resources

The BUILT-IN Help Pages
The built-in Help link in PowerSchool has good documentation on Interactive Reports.

KB 77143 - **Enterprise Reporting Frequently Asked Questions for PowerSchool 10.x**
https://support.powerschool.com/article/77143

KB 77492 - **How To - Setup and Access Enterprise Reporting (APEX) for PS 10.x**
https://support.powerschool.com/article/77492

Setting up Enterprise Reporting in PowerSchool 10.0
http://xcodeclub.org/ps/SettingUpEnterpriseReportingPowerSchool10.pdf
A better guide setting up Enterprise Reporting. Written by Dean B. Zaharis and the Maine PowerSchool Users Group.

ORACLE APEX Resources
http://apex.oracle.com

Oracle End User Interactive Report documentation
https://docs.oracle.com/cd/E59726_01/doc.50/e39146/int_rpt.htm#AEEUG453

Missing "Enterprise Reporting" tab due to legacy customization
https://www.sisresources.com/resources/latest-news/261-missing-tabs-on-the-system-reports-page
Blog post by Matt Freund with tips on how to resolve the missing tab issue some users are experiencing. If you are missing the Enterprise Reporting tab and it appears when you turn customizations off, check out this post.

North Carolina Enterprise Reporting Presentation
http://www.nc-sis.org/Documents/symp_2016/presentations/Enterprise_Reporting.pdf
Good presentation by NC, a bit dated because written for version 9, but the section on the options available when running reports is especially helpful.

Export an Interactive Report (Oracle documentation on exporting "Application Pages")
http://docs.oracle.com/cd/E59726_01/doc.50/e39147/deploy_export.htm#HTMDB25821

Import an Interactive Report (Oracle documentation on importing "Application Pages")
http://docs.oracle.com/cd/E59726_01/doc.50/e39147/deploy_import.htm#HTMDB25833

Concepts

Before you start, it's important to understand some "vocabulary".

First is "location".

1. The **Enterprise Reporting interface** that appears within PowerSchool. It's a tab on the System Reports page. This is where you and other admins will run reports. If you are a "manager", you can edit/delete/add reports to this page (if they exist in APEX).
   NOTE*: If an old customization is interfering with the tab, see this post by Matt Freund:*
   https://www.sisresources.com/resources/latest-news/261-missing-tabs-on-the-system-reports-page
2. The **APEX "Application Builder" (***also called the* ***"Enterprise Reporting Portal"*****)** interface that is completely outside of PowerSchool. This area is for creating additional reports and should be very limited access, only to your "developers".

Second is "Roles"

1. **Enterprise Reporting User** - Anyone who runs a report
2. **Enterprise Reporting Manager** - Can enable/disable reports and assign categories.
3. **APEX Report Builder** - Can create/import additional reports via the APEX portal.

Third is "Categories". Reports can be assigned to categories and access to those categories can be controlled using User Access Roles.

Fourth is "SQL". Enterprise Reports are based on "Structured Query Language". If you plan on developing your own reports or modifying existing reports, expect to learn at least a moderate amount about SQL and how to construct queries that extract data from PowerSchool.

Installation Steps

Be sure to read the "Setting up Enterprise Reporting in PowerSchool 10.0" by the Maine User Group. The steps below are broad outlines.

- Create Roles that can access ER.
- Add roles to users.
- Grant access to "APEX" for those who will be developing reports.

Using Enterprise Reports

PowerSchool uses the term "Enterprise Report" to refer to an "Interactive Report" from Oracle's APEX platform.

The following "pre-built" reports are included from PowerSchool.

- **Student Attendance -** data from the Students, Schools, FTE, Attendance and Attendance_Code tables.
- **YTD Absences by Grade** - a chart of absences by grade as well as a related data grid, which provides data from the Students, Schools, FTE, Attendance and Attendance_Code tables.
- **Student Incidents** - data from the Students, Schools, FTE and several Incident tables.
- **Student Grades** - data from the Students, Schools, FTE and StoredGrades tables.
- **Student Tests -** data from the Students, Schools, FTE, Test, TestScores and StudentTestScores tables.
- **Student Demographics** - data from the Students and Schools tables.

To run a report a user selects a report from the Enterprise Reporting tab. Even those with just a "User" level of access have a lot of flexibility when running a report. They can:

- Set Current Selection - to make the filtered results of the report the current selection of students in PowerSchool.
- Search the results of the report for a specific student or other criteria
- Add more filters to refine the report results.
- Add or remove columns
- Click on column headers for sorting and more filtering
- Group results with a "Control Break"
- Display a list of distinct values found in the columns
- Save a customized "version" of the report, just for themselves (private) or all (public)
- Access the "versions" of a report that they or others have saved
- Highlight data in the report based on certain criteria.
- Create basic charts
- Use advanced data summary features, such as "Compute", "Aggregate", "Group By", and "Pivot"
- Download the report in CSV or PDF format.
- Email the report and/or subscribe to the report (NOTE: currently some/all features of email and subscriptions are not working)
- Flashback (I HAVE USED THIS!). See your data as it was in the past, based on your server's backup and flashback settings. Can be useful for recovering data after an "accident". Default limit is 2 weeks, enter value in minutes.

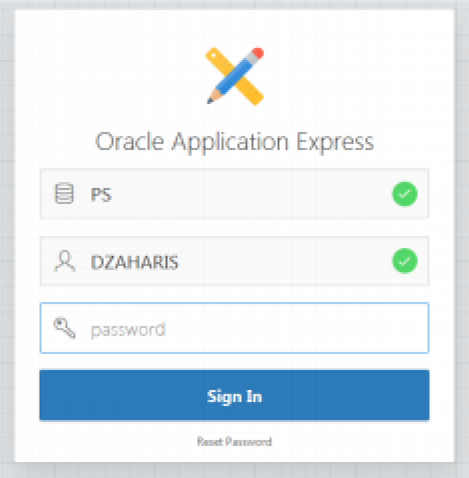
You may find a presentation from North Carolina useful. Some if it is based on the 9.2 preview release, but the sections on how to use the features of an Interactive Report are very helpful. http://www.nc-sis.org/Documents/symp_2016/presentations/Enterprise_Reporting.pdf

APEX Application Builder

The rest of this document is focused on the “Builder” area to help you create additional reports in APEX.

- Follow the steps in the Installation area above to enable a user to access the APEX Application Builder. (Be sure user has an email address setup and visit the "Applications" tab of their security settings).
- Use the address:
  https://[your powerschool server address]:8443/ords
- Tip for the correct URL, load any report, trim anything after ords/
   - for example, https://ps9.psugcal.org:8443/ords

    - Use PS for the Workspace field



PS

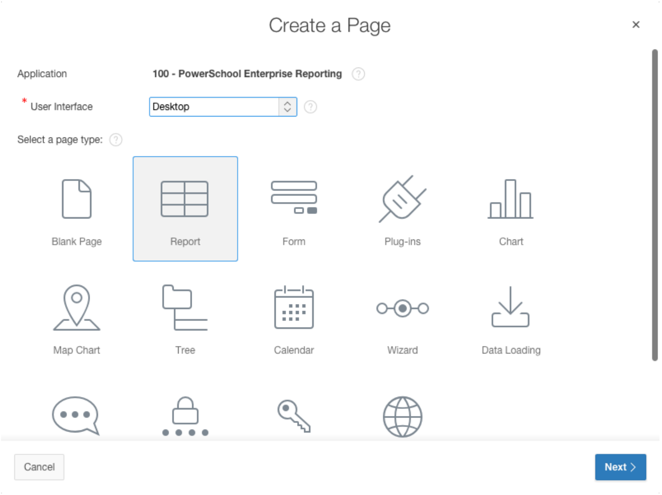
DZAHARIS

Tips:

- If there is no school filter applied to the report, the report may return more results than you're expecting. The Data Restriction Framework will implement a filter on the current school, but the results will include any student who has been or will be (based on "Next School" setting) at that school, including inactive students. If the user expects results similar to the main PowerSchool interface, which only shows active students at the currently selected school, you should add filters, perhaps as saved report, based on the School Name and Enroll_Status=0.
- You can leverage an automatic filter for YearID
    - Include the 2-digit YearID as Year (i.e. 26 for the 16-17 year) for an automatic initial filter for the current year. Here is an example of how to produce this in an sql query that includes the sections table:
        - round(SECTIONS.TERMID/100) as Year
- Create broad queries. Remember that your users can interact with the report and save their settings as variations of your initial report.
- Each APEX report is called a “page” and has a “page number”. If you plan on sharing your pages or copying them to other servers, be aware that importing a page with the same page number as one already on your server will overwrite the existing report. Number your pages with this in mind.
- You CAN export a page (report) and import it into another server, but keep the page number caveat above in mind.
- PSHTML is NOT available in APEX. Tags like ~(curstudid), ~(curyearid) or ~(curtermid) are not available. But remember to create your queries broadly and let your users add their own filters.

Creating an additional Page (Report) in APEX

- **Click on “Application Builder”**
- **There are 2 “PowerSchool Enterprise Reporting” Applications**
    - Application 100 is for users, put all your own reports here
    - Application 200 is for PowerSchool. Do not add/remove anything here, PowerSchool may add items in the future and may overwrite anything here.
- **Choose Application 100**
- **Click “Create Page”**
- **Select “Report” for a page type and click “Next”**



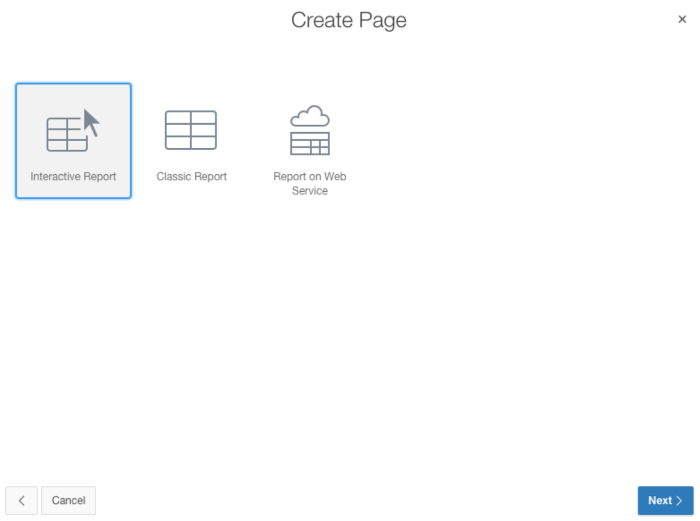
Create a Page

Application 100 - PowerSchool Enterprise Reporting

* User Interface Desktop

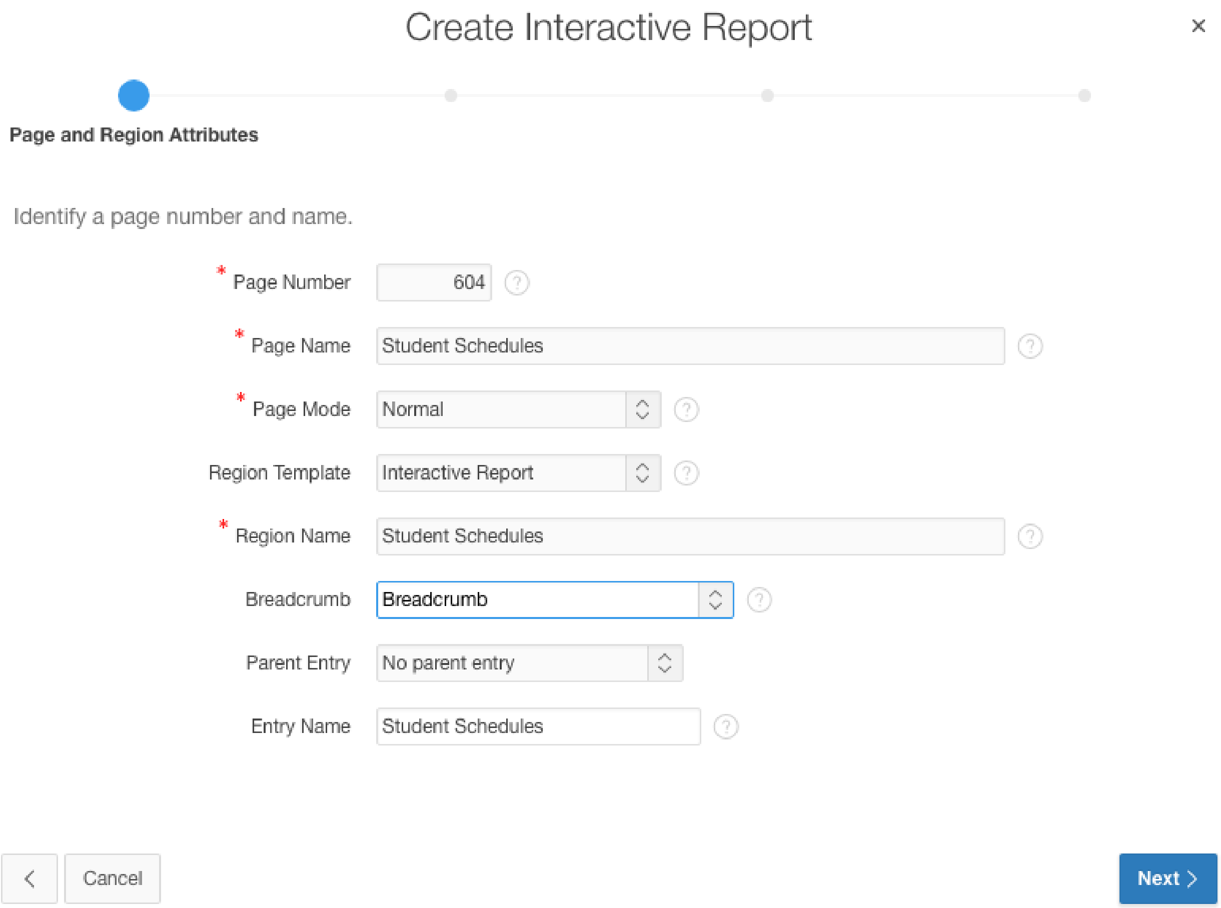
Select a page type:

- **Choose “Interactive Report” and click “Next”**



Create Page

- **CREATE INTERACTIVE REPORT**
  - Page Number = (devise a numbering system for yourself, i.e. 5xx)
  - Page Name = (A name of your choosing)
  - Page Mode = Normal
  - Region Template = Interactive Report
  - Region Name = (same as Page name)
  - Breadcrumb = Breadcrumb
  - Parent Entry = No parent entry
  - Entry Name = (same as Page name)



Create Interactive Report

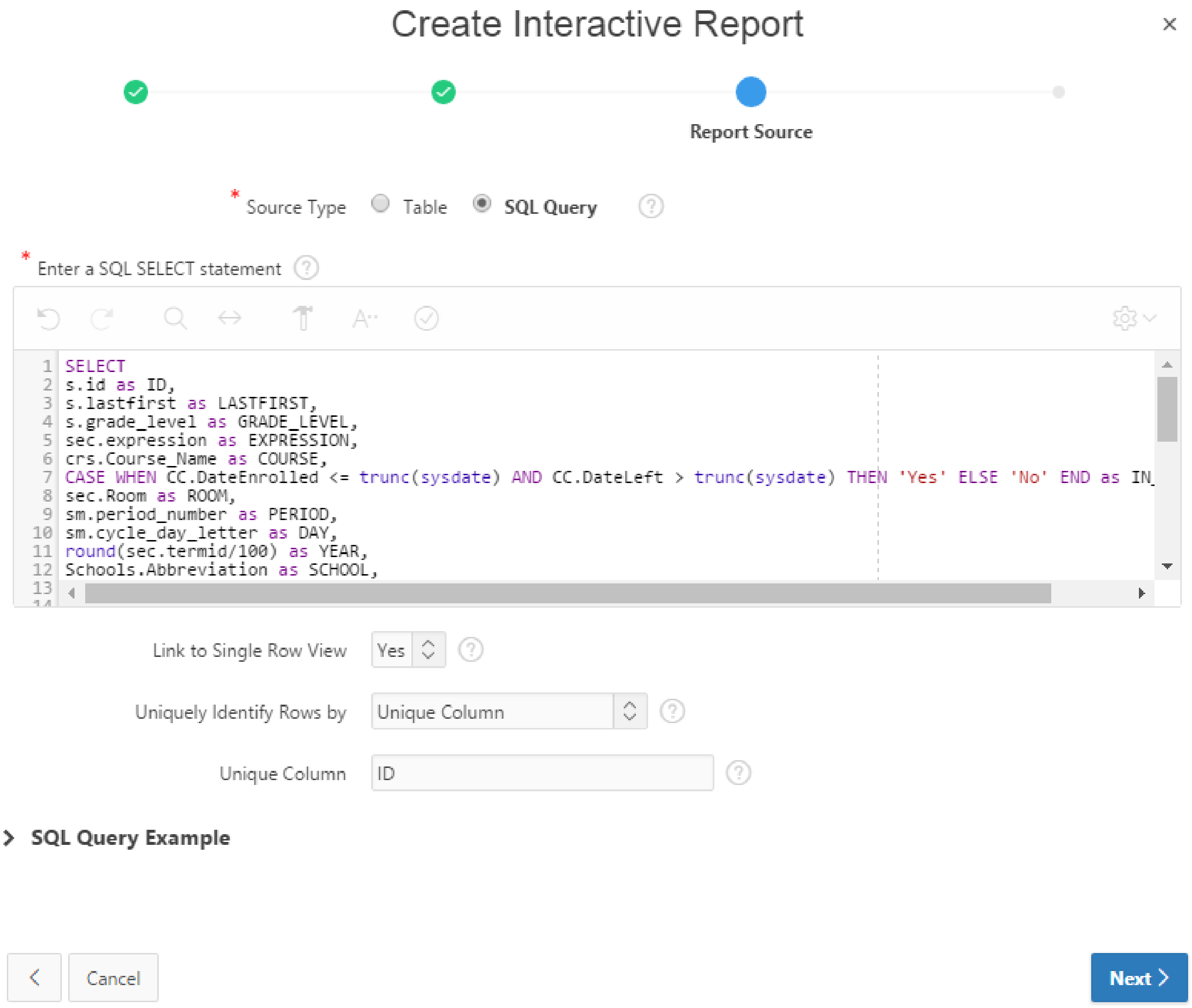
Page and Region Attributes

Identify a page number and name.

* Page Number	604
* Page Name	Student Schedules
* Page Mode	Normal
Region Template	Interactive Report
* Region Name	Student Schedules
Breadcrumb	Breadcrumb
Parent Entry	No parent entry
Entry Name	Student Schedules

- **NAVIGATION MENU**
  - Navigation Preference = Do not associate this page with a navigation menu entry (navigation is managed in PowerSchool) - click “Next”

- **REPORT SOURCE (SQL Query example with "Student Schedules")**
  - Source Type = SQL Query
  - Enter a SQL SELECT statement
    - Enter/Paste your query here.
    - You can click the "Hammer" icon to visually build a query, but it's better to have a working query you've debugged in another tool ready to paste
    - When writing query, consider aliases that have underscores where you want spaces, which will also capital case the result - i.e. TESTSCORE = Testscore but TEST_SCORE will render as Test Score. See alias examples in the "Student Schedules" query.
  - Link to Single Row View = Yes (if you want to allow a single record "form" view)
  - Uniquely Identify Rows by = Unique Column
    - The default of "ROWID" will often create an error with complex queries.
  - Unique Column = ID
    - In this example, ID is an alias for a column with the student's ID. This should be a column from your query that is a key field for one of your primary tables.



Create Interactive Report

* Source Type ○ Table ● SQL Query

* Enter a SQL SELECT statement

```
SELECT
s.id as ID,
s.lastfirst as LASTFIRST,
s.grade_level as GRADE_LEVEL,
sec.expression as EXPRESSION,
crs.Course_Name as COURSE,
CASE WHEN CC.DateEnrolled <= trunc(sysdate) AND CC.DateLeft > trunc(sysdate) THEN 'Yes' ELSE 'No' END as IN_
sec.Room as ROOM,
sm.period_number as PERIOD,
sm.cycle_day_letter as DAY,
round(sec.termid/100) as YEAR,
Schools.Abbreviation as SCHOOL,
```

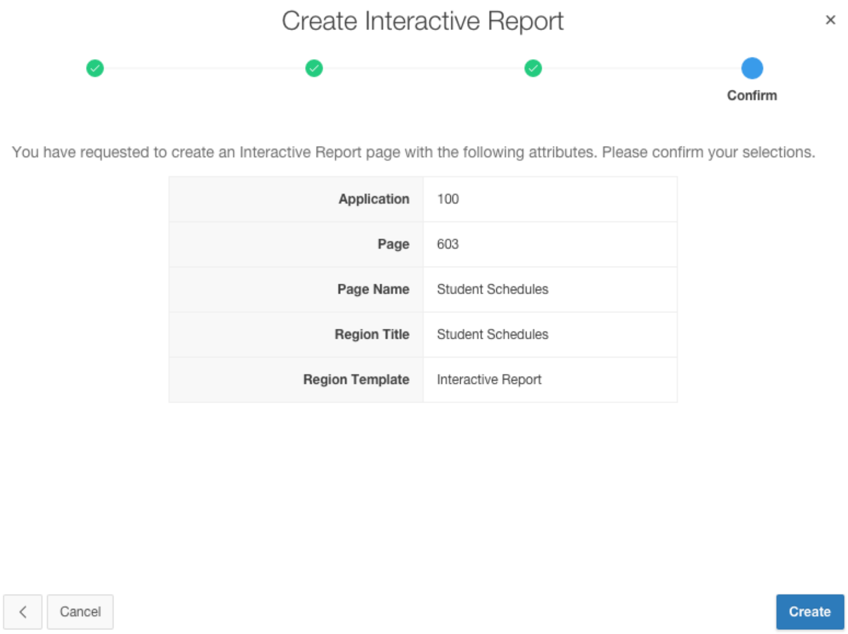
Link to Single Row View: Yes

Uniquely Identify Rows by: Unique Column

Unique Column: ID

CONFIRM

- ○ (Review details)
- ○ Create



Create Interactive Report

You have requested to create an Interactive Report page with the following attributes. Please confirm your selections.

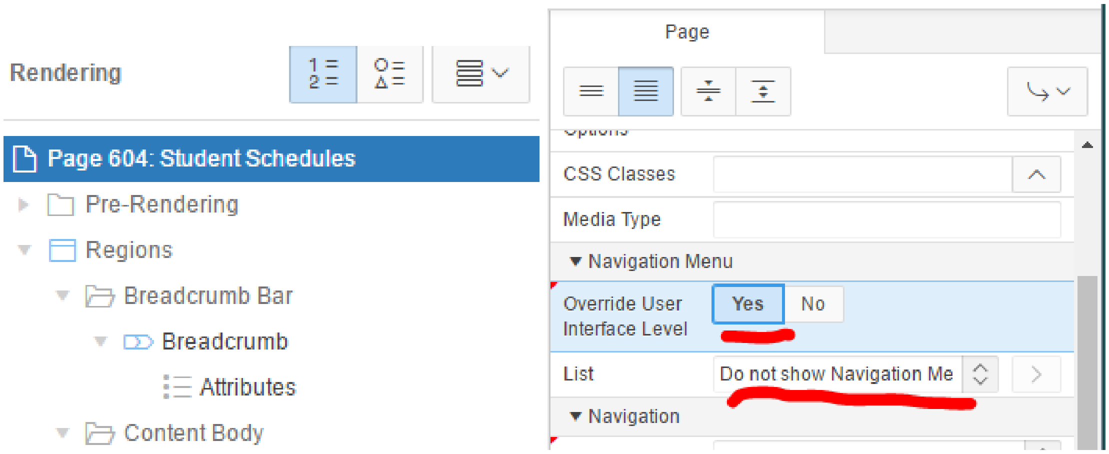
Application	100
Page	603
Page Name	Student Schedules
Region Title	Student Schedules
Region Template	Interactive Report

Additional Settings to update

Once your page is saved, click on it to enter the “Page Designer”. On the left side is a list of the different sections of your report page. When you select an area of the report, on the right side is the “Property Editor” with additional properties for that region of the page. Below are additional settings to consider when creating your own pages. (Credit to Jason Springel from PowerSource Forum Post https://support.powerschool.com/thread/17749)

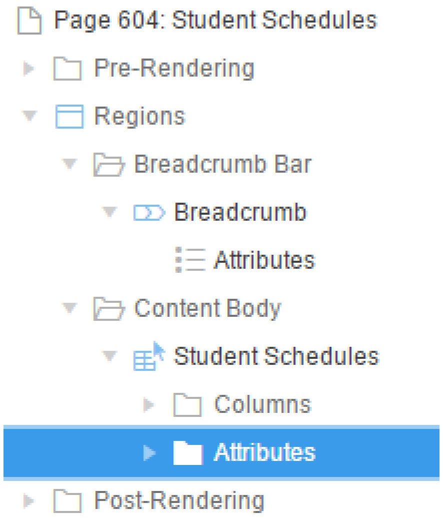
REMEMBER TO CLICK "SAVE" AT THE TOP AFTER CHANGING SETTINGS

- Select on the left **Page xxx:(Name)**
  - ○ Navigation Menu - Override User Interface Level from No to Yes.
  - ○ Keep the default setting for List, which is "Do not show Navigation Menu List".
  - ○ This will suppress a blue "left navigation" area we don't need for PowerSchool.

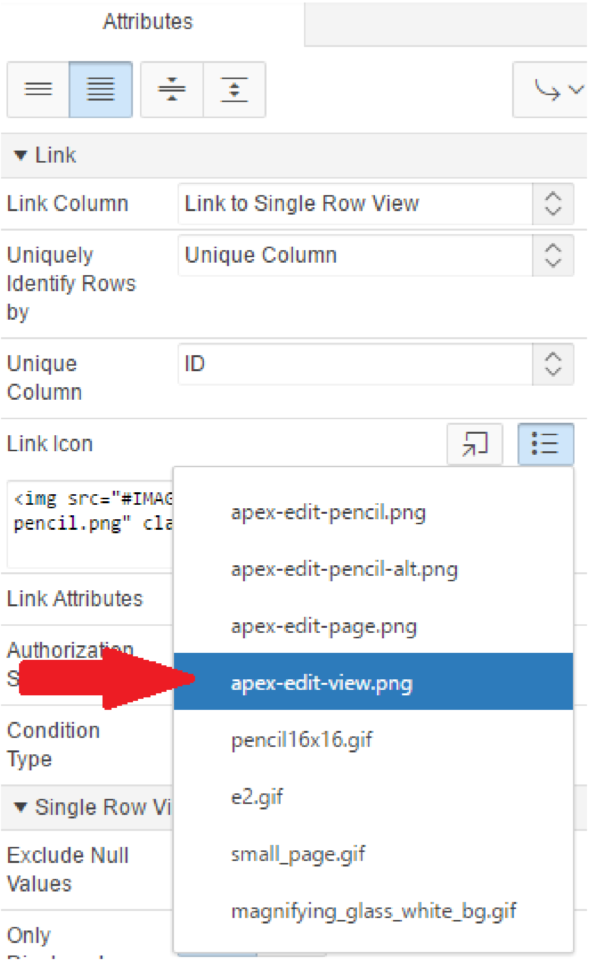


Override User Interface Level: Yes

List: Do not show Navigation Me



- Select on the left **Page ## > Regions > Content Body > (Report name) > Attributes** (icon looks like a folder)



Link Column: Link to Single Row View

Uniquely Identify Rows by: Unique Column

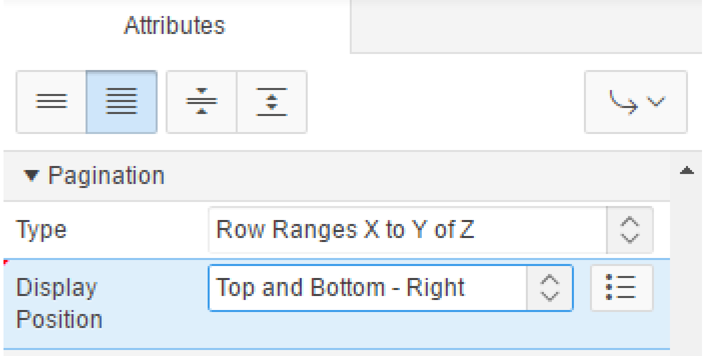
Unique Column: ID

Link Icon

```
<img src="#IMAG
pencil.png" cla
```

    ○ **Link Icon -** (If you enabled a Link Column). The icon default to a pencil, but since we are creating "view only" screens, we should change it to a magnifying glass. From the "Quick Pick" icon, select "apex-edit-view.png"

- **Pagination - Type**: Set to "Row Ranges X to Y of Z" so that you can see the total number of records.
- **Pagination - Display Position**: The default is Bottom - Right, but you may want to consider "Top and Bottom - Right". This is subjective, but this setting allows users to see the pagination information at the top and bottom, so they don't have to scroll past the displayed rows to see this information. Try it and see what you think.

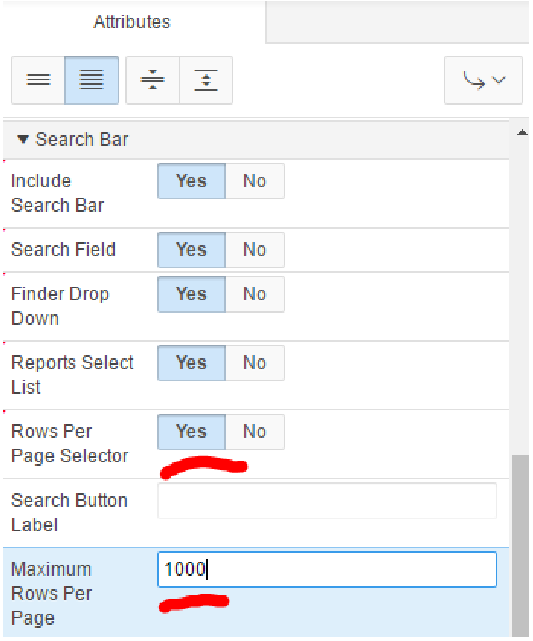


▼ Pagination

Type: Row Ranges X to Y of Z

Display Position: Top and Bottom - Right

- **Search Bar - Rows Per Page Selector**: Change from No to Yes.
- **Search Bar - Maximum Rows Per Page**: Set to "1000" as a reasonable limit on how many rows to show on one page. This will prevent overload of data from crashing the web page.



▼ Search Bar

Include Search Bar: Yes No

Search Field: Yes No

Finder Drop Down: Yes No

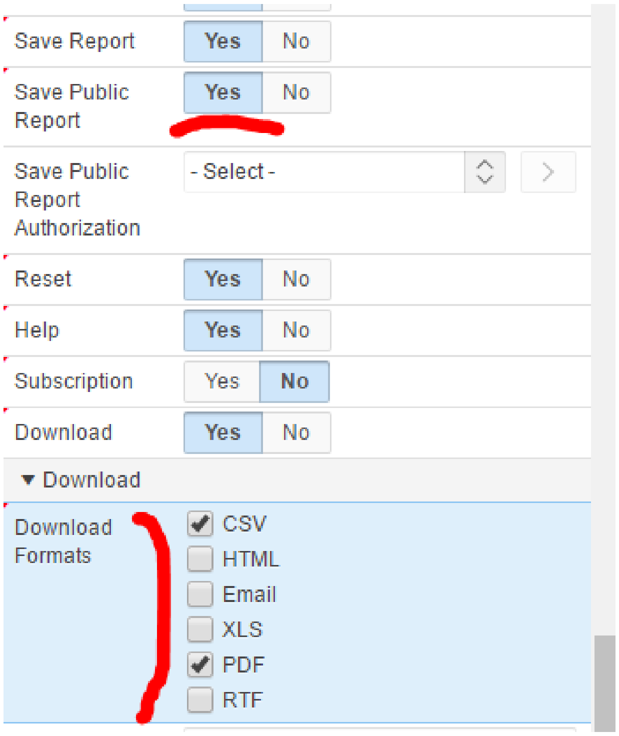
Reports Select List: Yes No

Rows Per Page Selector: Yes No

Search Button Label:

Maximum Rows Per Page: 1000

- **Actions Menu - Save Public Report**: Decide if you want to change this from No to Yes. This will allow your users to save a version of the report and make it "public" so others can see. If not checked they can only save reports for themselves.
- **Download**: Uncheck the boxes for HTML, Email, and XLS. Leave just CSV and PDF checked, if those two options are in line with the options you wish to offer. *Email is not recommended as of 10.0.3 due to unresolved issues with emailing in APEX.*


Save Report: Yes No

Save Public Report: Yes No

Save Public Report Authorization: - Select -

Reset: Yes No

Help: Yes No

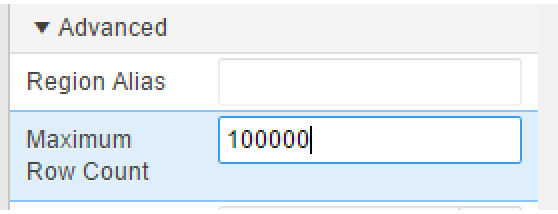
Subscription: Yes No

Download: Yes No

▼ Download

Download Formats:
☑ CSV
☐ HTML
☐ Email
☐ XLS
☑ PDF
☐ RTF

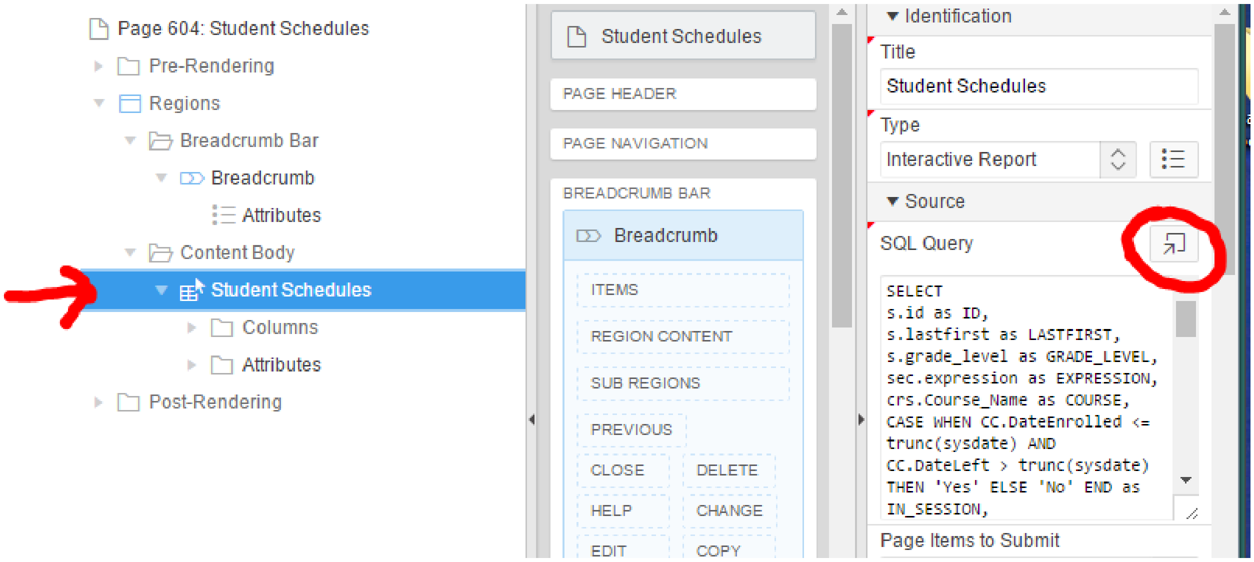
- **Advanced - Maximum Row Count**: This is a very important setting and can prevent system performance issues. The default is '1000000' or one million rows. PowerSchool's report defaults are 100,000 or '100000'. Depending on the data, it may make sense to lower this number even further, say '50000'.



Region Alias:

Maximum Row Count: 100000

- **Under Content Body select the name of your Interactive Report Region**
    - The SQL Query can be modified here, if necessary. Changing column aliases will not always have the same effect as when first entered in the wizard. For example, `students.grade_level as GRADE_LEVEL` will result in the column header being "Grade level" and not "Grade Level". For this reason it's recommended to have the query ready for the page creation wizard.



Title: Student Schedules

Type: Interactive Report

SQL Query:

```
SELECT
s.id as ID,
s.lastfirst as LASTFIRST,
s.grade_level as GRADE_LEVEL,
sec.expression as EXPRESSION,
crs.Course_Name as COURSE,
CASE WHEN CC.DateEnrolled <=
trunc(sysdate) AND
CC.DateLeft > trunc(sysdate)
THEN 'Yes' ELSE 'No' END as
IN_SESSION,
```

- **REMEMBER TO CLICK "SAVE"** at the top after changing your settings.

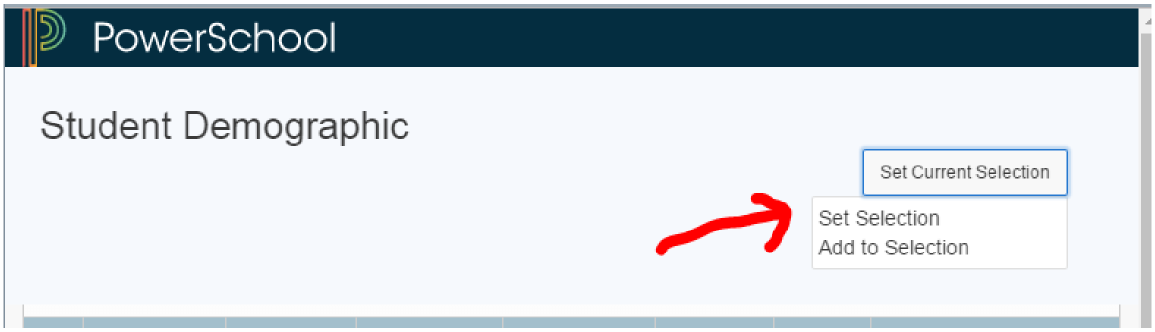
Adding your report to the list

Return to the Enterprise Reporting interface inside of PowerSchool. If a user is assigned a role that is allowed to "Manage Enterprise Reports", they can manage this list. If the user has the manager role, there will be an "Add" button at the top right -- when clicked:

- Choose an available Report Name from the available list.
- Enter a **description** for the report and choose a report **category**.
- **State**: Select one of the following options:
    - **Published** to publish the report, so that the report will be seen by users that have been granted access to that category.
    - **Not Published** to not publish the report, so that the report will only be seen by users that have been given permission to manage Enterprise Reports.

Implementing the "Set Current Selection" button

PowerSchool's built-in reports feature a "Set Current Selection" button. If the report lists students, the user can run the report, apply filters, and make the results the current selection or add the results to a current selection. Here's how to add that feature to your reports.
NOTE: ***As of version 10.0.0.3 there is a bug where the "Current Selection" feature will return an error if the user does not have the Enterprise Reporting "Manager" role.***



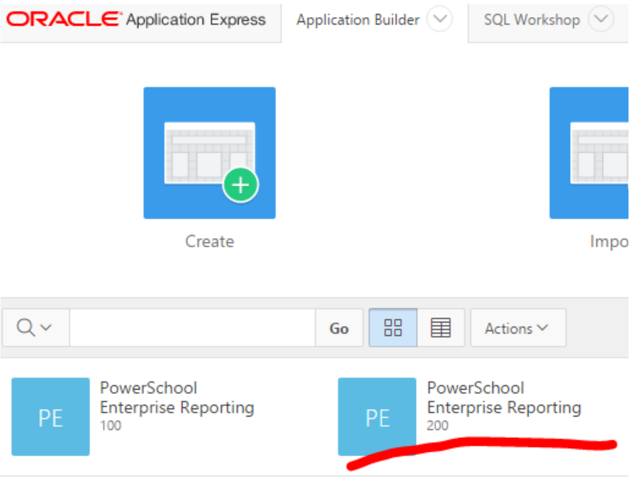
Requirements:

- Your query must include a column for the student ID with an alias of "ID".
- Your query must include a column for the student's lastfirst field, with an alias of "LastFirst".
    - Note: If you are querying the fields directly from the students table, it is not necessary to alias them, but it doesn't hurt.
    - For example:

SELECT s.id, s.lastfirst, etc... FROM students s	OR	SELECT sg.studentid as ID, s.lastfirst as LASTFIRST etc… FROM storedgrades sg INNER JOIN students s ON sg.studentid = s.id

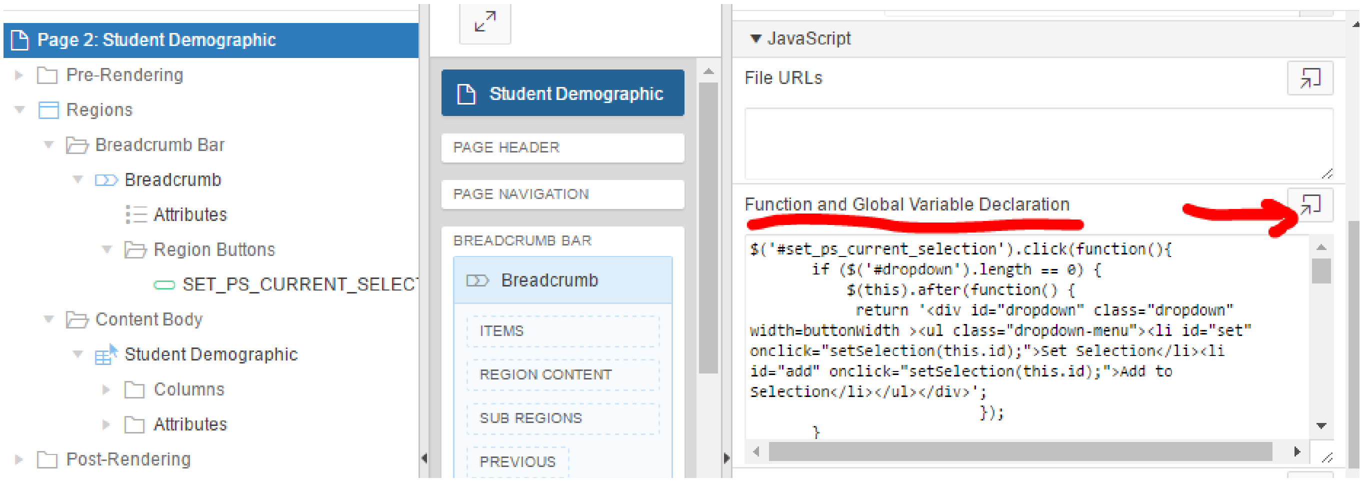
Copy Script and CSS from a built-in report

Go back to the APEX home page (I find it helpful to right-click on the "ORACLE Application Express" link at the top and open in a new tab so I can have two tabs). Then click on "Application Builder" and then on "PowerSchool Enterprise Reporting 200" (the application where all the built-in reports live).


ORACLE Application Express
Application Builder
SQL Workshop
Create
Impo
Go
Actions
PE
PowerSchool Enterprise Reporting
100
PE
PowerSchool Enterprise Reporting
200

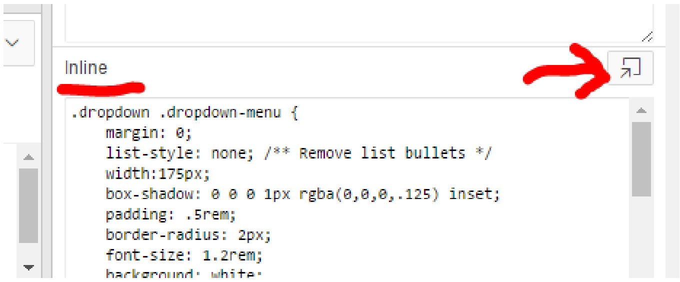
- **Open the "Student Demographic" report** (or any other built-in report with the button)

- Select on the left **Page 2: Student Demographic.** Then on the right scroll down the JavaScript and CSS section and copy and paste the contents of the following attributes to the same attributes in your report. I usually click on the "Code Editor" button to make sure I can see and copy the entire contents:
    - JavaScript -> **Function and Global Variable Declaration**
    - CSS -> **Inline**



Function and Global Variable Declaration

```
$('#set_ps_current_selection').click(function(){
       if ($('#dropdown').length == 0) {
           $(this).after(function() {
            return '<div id="dropdown" class="dropdown"
width=buttonWidth ><ul class="dropdown-menu"><li id="set"
onclick="setSelection(this.id);">Set Selection</li><li
id="add" onclick="setSelection(this.id);">Add to
Selection</li></ul></div>';
                          });
       }
```

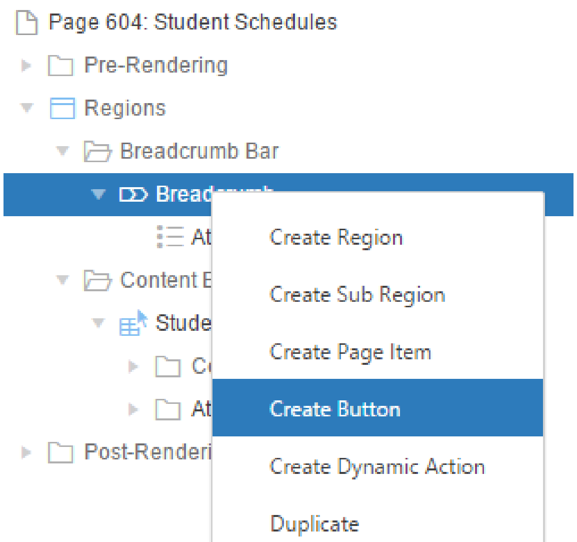


Inline

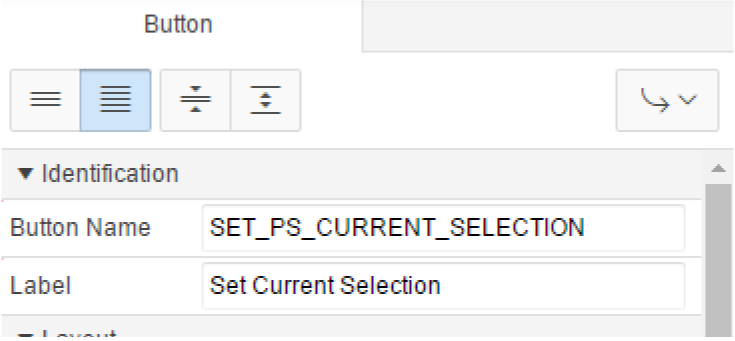
```
.dropdown .dropdown-menu {
    margin: 0;
    list-style: none; /** Remove list bullets */
    width:175px;
    box-shadow: 0 0 0 1px rgba(0,0,0,.125) inset;
    padding: .5rem;
    border-radius: 2px;
    font-size: 1.2rem;
    background: white;
```

Create the "Set Current Selection" button

- Back in your new report, right-click on "Breadcrumb" and choose "Create Button"



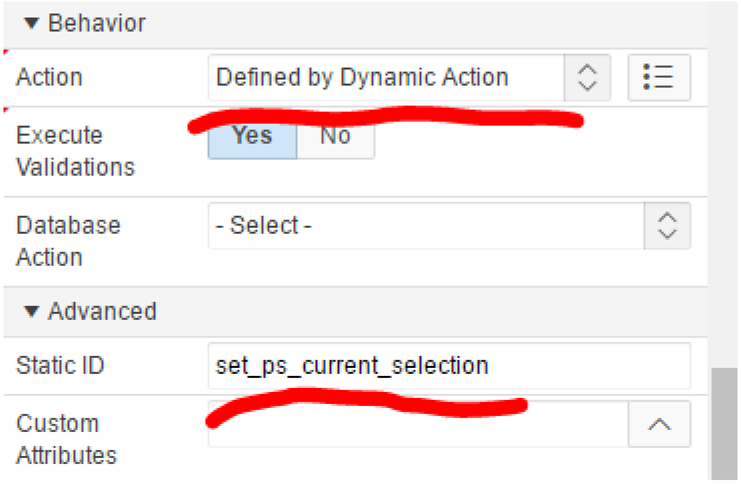
- To the right, complete the following in the "Identification" section:
  - **Button Name**: SET_PS_CURRENT_SELECTION
  - **Label**: Set Current Selection



▼ Identification

Button Name	SET_PS_CURRENT_SELECTION
Label	Set Current Selection

- Scroll down to "Behavior" and change the following:
  - **Action:** Defined by Dynamic Action
- Under "Advanced" and set the following:
  - **Static ID:** set_ps_current_selection



▼ Behavior

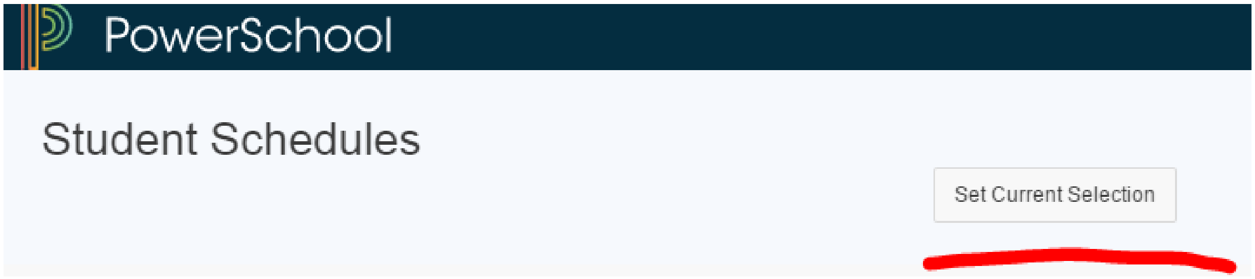
Action	Defined by Dynamic Action
Execute Validations	[✓ Yes] No
Database Action	- Select -

▼ Advanced

Static ID	set_ps_current_selection
Custom Attributes	

REMEMBER TO SAVE!

Once you click "Save" at the top, try running your report and test the button.



Student Schedules

SQL EXAMPLES

See psugcal.org for the full import files that you can import with the “Import/Export” area of APEX Application Builder. Here are some sql examples for additional reports.

Room Utilization

This report works especially well when using the Pivot feature of Interactive Reports

```
SELECT
schools.name as SCHOOL,
sections.room as ROOM,
sections.expression as EXPRESSION,
section_meeting.period_number as PERIOD,
section_meeting.cycle_day_letter as DAY,
sections.termid as TERMID,
round(sections.termid/100) as YEAR,
sections.no_of_students as NUM_STUDENTS

FROM
sections
INNER JOIN schools
      ON sections.schoolid = schools.school_number
INNER JOIN section_meeting
      ON sections.id=section_meeting.sectionid
```

Student Schedules

```
SELECT
s.id as ID,
s.lastfirst as LASTFIRST,
s.grade_level as GRADE_LEVEL,
sec.expression as EXPRESSION,
crs.Course_Name as COURSE,
CASE WHEN CC.DateEnrolled <= trunc(sysdate) AND CC.DateLeft >
trunc(sysdate) THEN 'Yes' ELSE 'No' END as IN_SESSION,
sec.Room as ROOM,
sm.period_number as PERIOD,
sm.cycle_day_letter as DAY,
round(sec.termid/100) as YEAR,
Schools.Abbreviation as SCHOOL,
s.enroll_status as ENROLL_STATUS,
t.lastfirst as TEACHER,
sec.Termid as TERMID,
Terms.Abbreviation as TERM,
CC.DateEnrolled as DATE_ENROLLED,
CC.DateLeft as DATE_LEFT

FROM
Students s INNER JOIN
CC ON s.id = cc.studentid INNER JOIN
Sections sec ON cc.sectionid = sec.id INNER JOIN
Section_Meeting sm ON sec.id = sm.sectionid INNER JOIN
Schools ON sec.schoolid = Schools.school_number INNER JOIN
Courses crs ON UPPER(sec.course_number) = UPPER(crs.course_number)
INNER JOIN
Teachers t ON sec.Teacher = t.id INNER JOIN
Terms ON sec.termid = Terms.ID AND sec.schoolid = Terms.SchoolID

ORDER BY
s.lastfirst,
sec.termid,
sec.expression
```

Teacher Percent Averages

This query filters to the current year only, based on the current date. It’s a good example of doing this without the access to PSHTML

```
SELECT
schools.name AS School,
sec.course_number,
crs.course_name,
sec.termid AS TermID,
tch.lastfirst AS Teacher,
pgf.finalgradename AS FinalGrade,
COUNT(pgf.percent) AS NumStudents,
ROUND(AVG(pgf.percent),1) AS PercentAvg

FROM
PGFinalGrades pgf
INNER JOIN Sections sec ON pgf.sectionid = sec.id
INNER JOIN cc ON pgf.sectionid = cc.sectionid AND pgf.studentid =
cc.studentid
INNER JOIN Teachers tch ON sec.teacher = tch.id
INNER JOIN Courses crs ON sec.course_number = crs.course_number
INNER JOIN terms t on sec.termid = t.id and sec.schoolid = t.schoolid
INNER JOIN terms t2 ON
 t2.schoolid = t.schoolid AND
 t2.firstday < t.lastday and t2.lastday > t.firstday AND
 trunc(sysdate) between t2.firstday and t2.lastday AND
 t2.isyearrec=1
INNER JOIN Schools ON sec.schoolid = Schools.school_number

GROUP BY
schools.name,
sec.course_number,
crs.course_name,
sec.termid,
tch.lastfirst,
pgf.finalgradename

ORDER BY
crs.course_name,
tch.lastfirst
```