PowerSchool Enterprise Reporting

Presented by Roger Sprik - June, 2020

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".

Enterprise Reporting is very powerful, very flexible, and customizing the reports is very visual. Some key features include:
- Saved Reports: The same Enterprise Report can be modified by users in different ways and that "version" of the report can be saved just for that user or for all users
- Flashback. See data in the recent past
- Flexibility. You can add your own reports or modify the included ones.

New in in APEX 5.1 with Oracle 12.2 (Fall 2019)

- Two-Pane mode
- Properties Filter to help find settings more quickly
- New Jet Charts
- New Interactive Grid report style with access to new Jet Charts

Resources

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

KB 77492 - **How To - Setup and Access Enterprise Reporting (APEX) for PS 10.x**
https://help.powerschool.com/t5/PowerSchool-SIS-Knowledge-Base/How-To-Setup-and-Access-Enterprise-Reporting-APEX-for-PS-10-x/ta-p/14376

PowerSchool Insider Episode 2 - **Enterprise Reports** (*How to use them*)
PowerSchool Insider Episode 56 - **Building Enterprise Reports**
https://support.powerschool.com/d/powerschool_insider_index

What's New in 9.2 (Beginning on Slide 24) - **Using Enterprise Reports**
https://support.powerschool.com/exchange/view.action?download.id=786

APEX - Top 5 Changes to Make - Tweaks for reports you create yourself
https://support.powerschool.com/exchange/view.action?download.id=875

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

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).
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. As of the writing of this document, it is not possible to add or modify the category list.

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.

Setup Steps

There are 2 or 3 types of APEX/Enterprise Reporting users.
- **Users**: "Access Enterprise Reports"
  - These are your end users that can run Enterprise Reports
- **Managers:** "Manage Enterprise Reports"
  - Can Manage the list of reports that appear on the Enterprise Reporting Tab, as long as those reports exist in the APEX portal.
- **Builders:** Access to the APEX Portal to build reports

Setup Roles for users to access and/or manage the Enterprise Reporting Tab

- Navigate to: Start Page > System > Roles Administration > User Access
- Click the name of a role that should have APEX permissions
- Select the Enterprise Reporting tab
- If users with this role should be able Manage Enterprise Reports (Create, Edit, Delete, etc), click Manage Enterprise Reports
- If the user should only be able to view and run Enterprise Reports, uncheck Manage Enterprise Reports and check the Report Categories they should have access to.
- Click Save
- Repeat steps 3-8 for any other Roles that need APEX permissions.

Report Builders: Grant Access to Enterprise Reporting Portal

A report builder has access to the "external" APEX Reporting Portal to be able to create, import, export, or update Enterprise Reports. This type of access should only be granted to a few select users.

- This is done on a user-by-user basis
- From staff search fine the user you want to have access to Enterprise Reporting
- Click Security Settings
- Select the Applications tab
- Check Oracle Application Express Account (APEX). The APEX Settings fields will appear.
- APEX portal passwords are managed separately from the main PS password
- Enter a temporary password. The user will be prompted to create their own on first logon.
  Note: Oracle password complexity requirements apply. The password must contain the following at a **minimum**
    - 1 or more Numbers
    - 1 or more Lowercase Letters
    - 1 or more Uppercase Letters
    - 1 or more of the following Special Characters **+,-,_,#,%** and no others
    - 8 total characters or more
- Click Submit

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.
- Additional reports that are available as of 19.11 (must be added to list by a manager)
    - **Current Grades - New**
    - **Contact Relationship Details - New**
    - **Course List - New**
   - **Student Demographic Plus - New**

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.
- 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". Amount of history available varies, but can be limited to hours or even minutes depending on how active the server is. Enter value in minutes.

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 Setup area above to enable a user to access the APEX Application Builder.
- 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://pstest.psugcal.org:8443/ords
        - Use PS for the Workspace field
        - If this is first time logging in, you will be prompted to change the password, but can keep the same one entered in setup
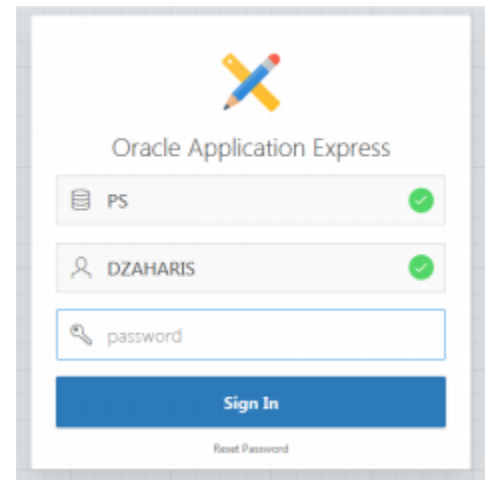
Tips:

- 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 a 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"**
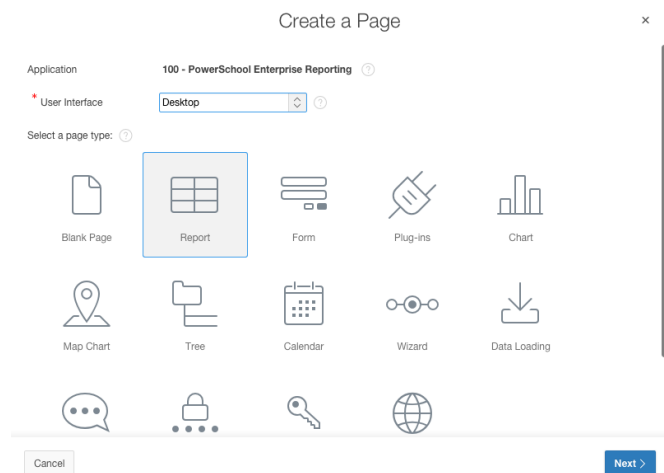
- **Choose "Interactive Report" and click "Next"**
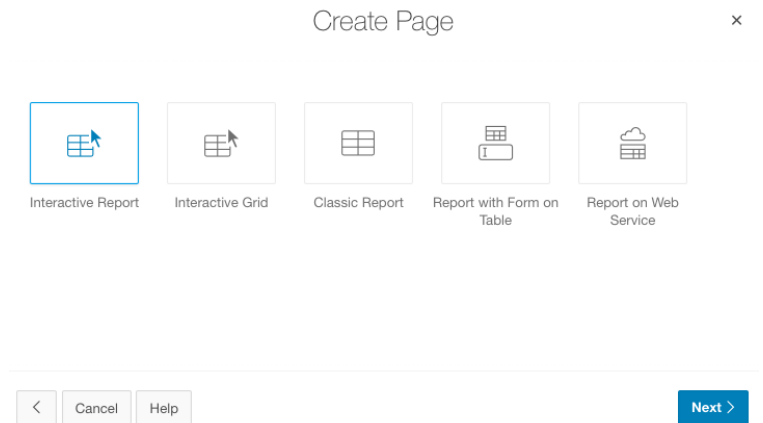
- **CREATE INTERACTIVE REPORT**
    - Page Number = (devise a numbering system for yourself, i.e. 5xx) - try to use a numbering scheme that will minimize conflicts with other reports
    - Page Name = (A name of your choosing)
    - Page Mode = Normal
    - Breadcrumb = Breadcrumb
    - Parent Entry = No parent entry
    - Entry Name = (same as Page name)

Create Interactive Report   ✕

Page Attributes

Type	**Interactive Report**
* Page Number	801 ⑦
* Page Name	Sample Report ⑦
Page Mode	**Normal** Modal Dialog ⑦
Breadcrumb	Breadcrumb ⌄ ⑦
Parent Entry	No parent entry ⌄
Entry Name	Sample Report ⑦

‹   Cancel   Next ›

- **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.
  - Click "Create"

Create Interactive Report

Report Source

* Source Type   ○ Table   ● SQL Query   ?

* Enter a SQL SELECT statement ?

Validation successful

```
1 SELECT
2 s.id as ID,
3 s.lastfirst as LASTFIRST,
4 s.grade_level as GRADE_LEVEL,
5 sec.expression as EXPRESSION,
6 crs.Course_Name as COURSE,
7 CASE WHEN CC.DateEnrolled <= trunc(sysdate) AND CC.DateLeft > trunc(sysdate) THEN
8
```

> SQL Query Example

< Cancel   Create

Additional Settings to update

Once your page is created, you will be taken to 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.

The center section will not be of much use. New in APEX 5.1 is the ability to switch to a 2-pane view from the "Settings" icon
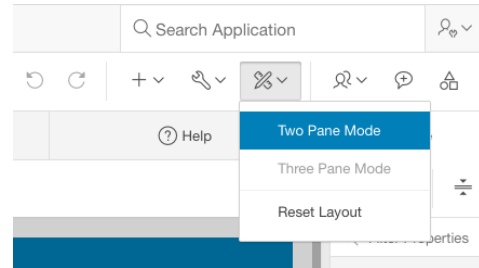
There are some additional changes you should make to make your new report look like the other PowerSchool reports.See the "**APEX - Top 5 Changes to Make**" document linked in the Resources setting. Be sure to follow that document to:

- Remove large blue box (empty left navigation menu)
- Modify pagination settings to look better for users
- Fix setting for the Search Bar
- Should users be able to save public reports?
- **Critical**: Set the maximum number of rows returned

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

Additional Changes to Consider

- For date fields adjust date format mask
    - Page #:Name -> Regions -> Content Body -> Report Name -> Columns
    - Highlight a Date column
    - Appearance section on right: Format Mask
        - For North America the drop-down may not have what you are looking for
        - Try "MM/DD/YYYY"
- Can also change header column labels in this section

Help

- About the only useful thing in the center pane is the "Help" tab, which provides contextual help for the property you are working on. If you have switched to "Two Pane" view, you can still access the help from the "question mark" tab.
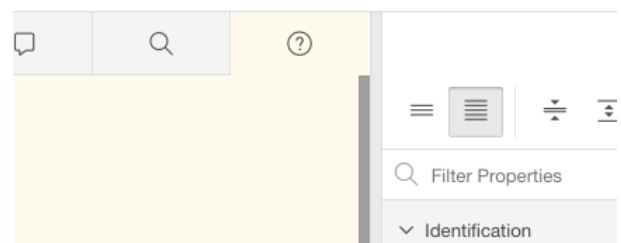
Editing a report

- To revisit and update a report, in the App Builder visit Application 100
- Remember that Application 200 contains the stock PS reports and editing those is NOT recommended. PowerSchool reserves the right to update those reports in the future and would overwrite any changes you make there.
  - Consider making a copy of a stock report into Application 100 via Export/Import if you would like to make modifications.

- **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. However, you can use the Columns area to update the column headers.
    - Adding columns to a query will automatically add those columns to all saved reports, including the Default View
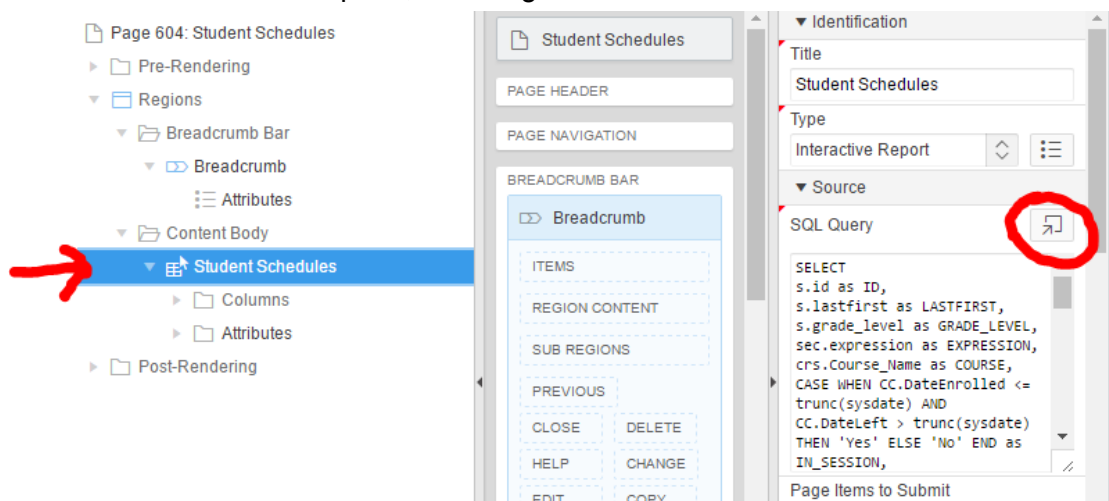
- **REMEMBER TO CLICK "SAVE"** at the top after changing your settings.

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.

Define a Default View

- Users can modify the report but as a report designer you should define a default view.
- Run the report from the Enterprise Reporting tab in PowerSchool
- Make the modifications necessary for the default view, for example:
  - Select only the columns most users would expect for a basic version of the report
  - Move the columns into a more logical order
  - Apply any expected default filters users might expect, such as student active enroll status
  - Apply any initial default sorting
  - Any other modifications desired
- When you have the report in a state that can be considered "Default", choose "Save Report and choose the "As Default Report" option.
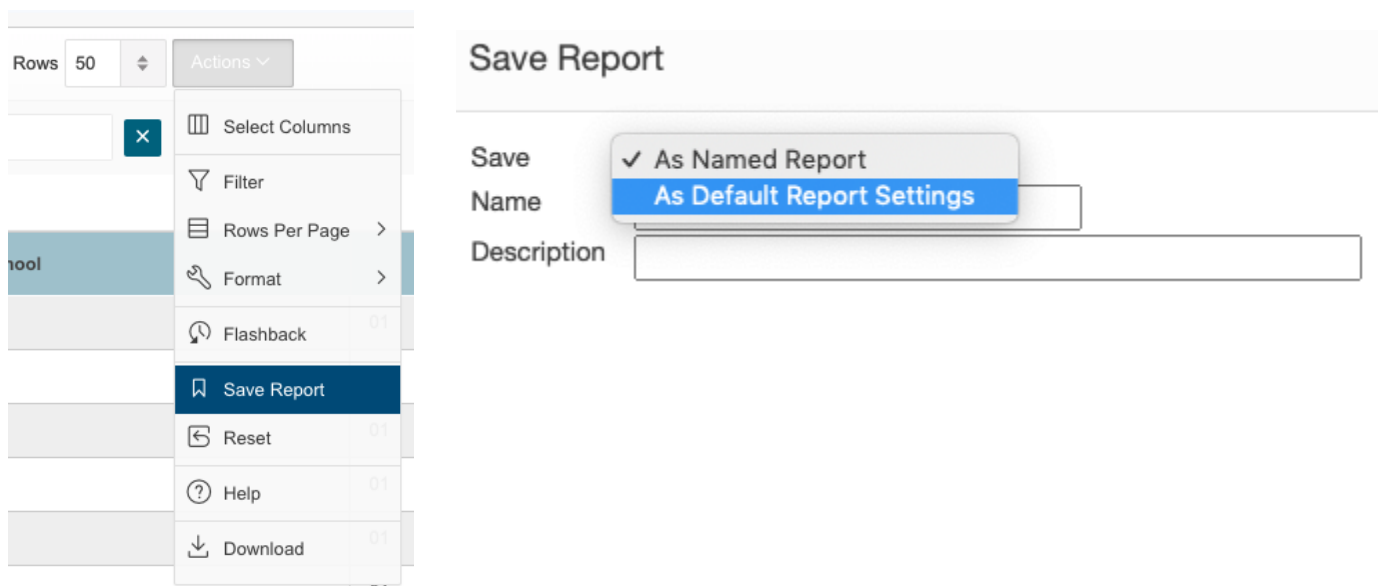
Delete a Report

- In Page Builder the Delete Page button is _____ on top ("wrench" icon)
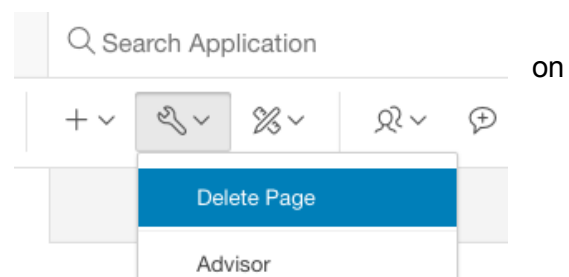
Import/Export to share reports

- Available shared reports to import
    - psugcal.org (Content Directory -> Enterprise Reporting) - unzip the download first
    - PowerSource Community Exchange
- To Import, within Application 100, choose "Export/Import" and then "Import"
    - Import file must end in .sql
- Export by first loading the page, then "Export" from the wrench icon

Inspecting Data and testing SQL Queries in APEX

- Apex Home > SQL Workshop
    - The SQL Workshop has a couple of tools that can help you learn the database schema, inspect data and test queries directly against your data.
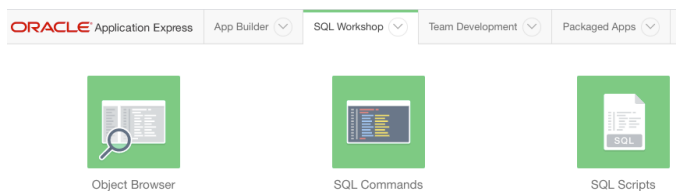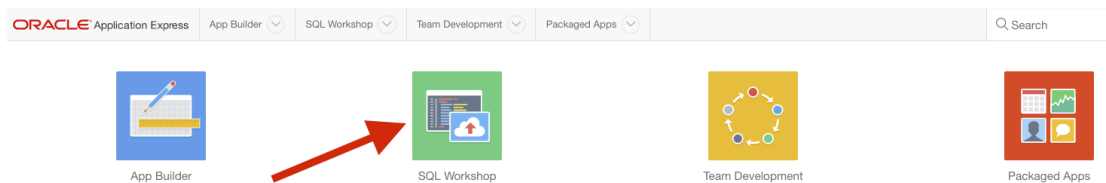
- **Object Browser** - with this tool you can inspect the database tables and fields
    - Tables will be blank, this is by design for security purposes
    - Use "Views" (which are representations of the tables)
    - Can view the data as it exists in Oracle (which can be different than what is shown in DDE/DDA)
- **SQL Commands** - write live queries directly against the database

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
```

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 may return an error if the user does not have the Enterprise Reporting "Manager" role.*
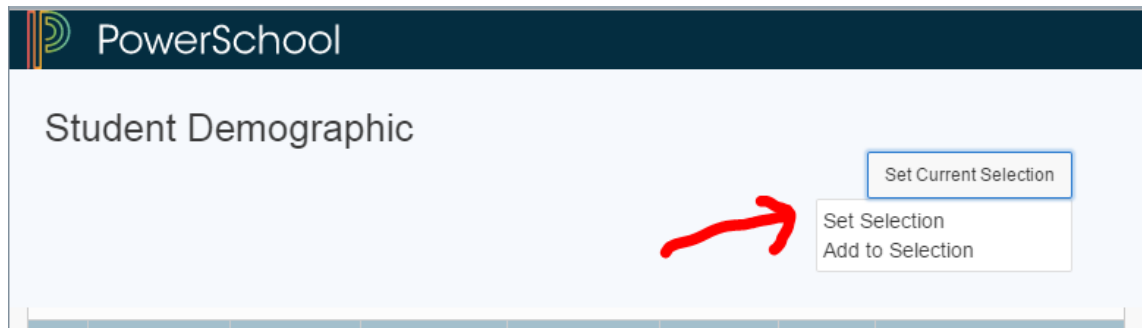

Requirements:
- (For **Students**) Your query must include a column for the student ID with an alias of "ID".
- (For **Students**) Your query must include a column for the student's lastfirst field, with an alias of "LastFirst".
    - Note: If you are querying the fields directly from the students table, 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

- (**For other tables**) Your query must include a column for the dcid of the table you wish to make a current selection in

Copy Script and CSS from a built-in report

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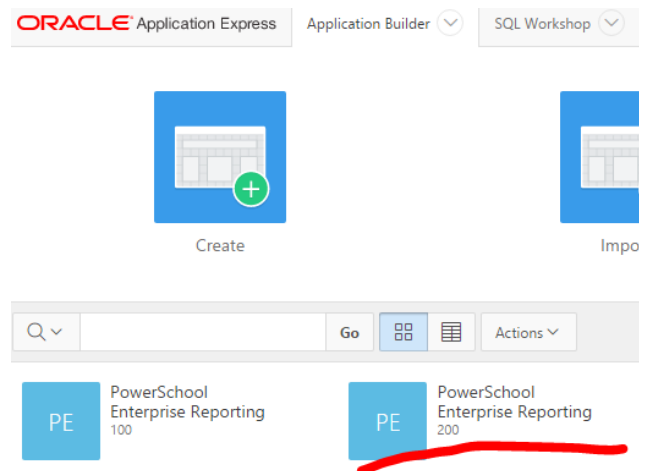
- **Open the "Student Demographic" report** (or any other built-in report with the button)

- 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**
- **For other tables:** Find a report and copy the Javascript and CSS code from the non-student table report. You will need to modify the table number and table name for your new report.
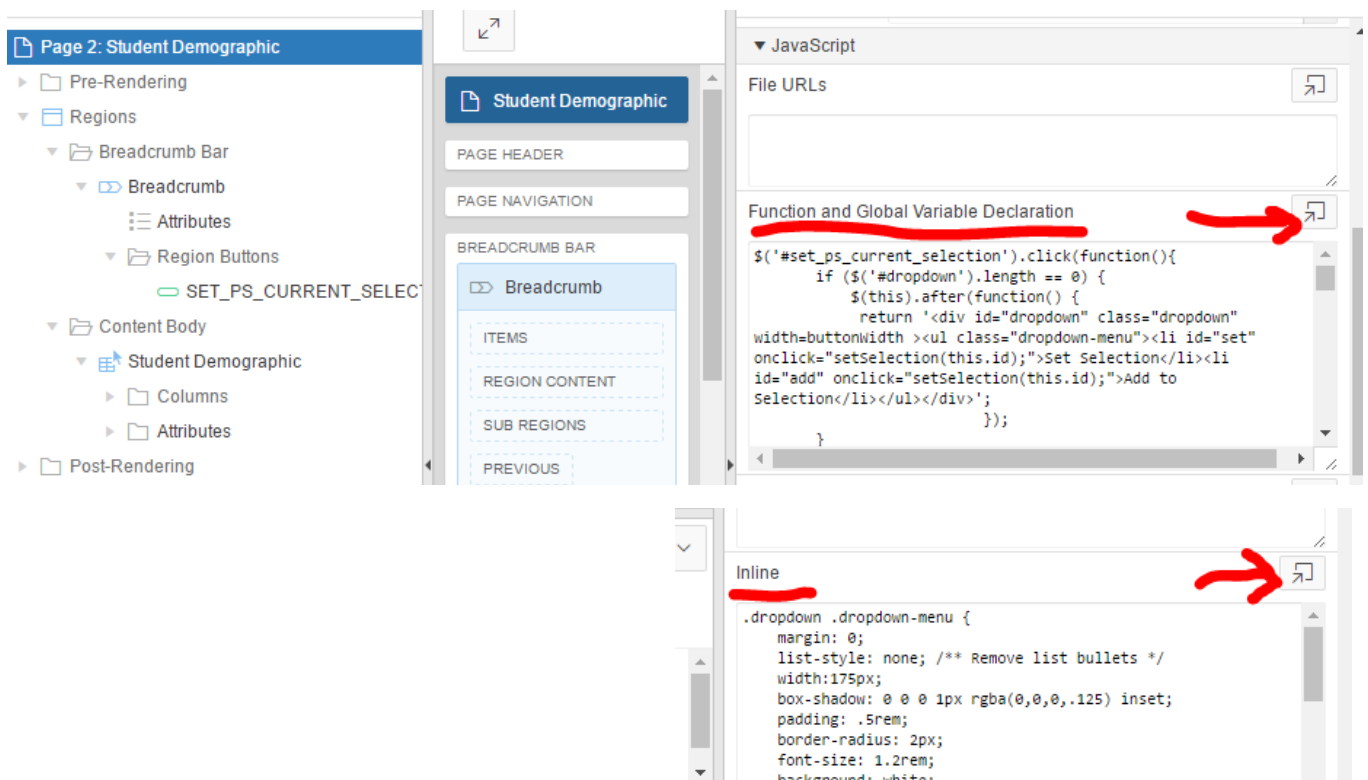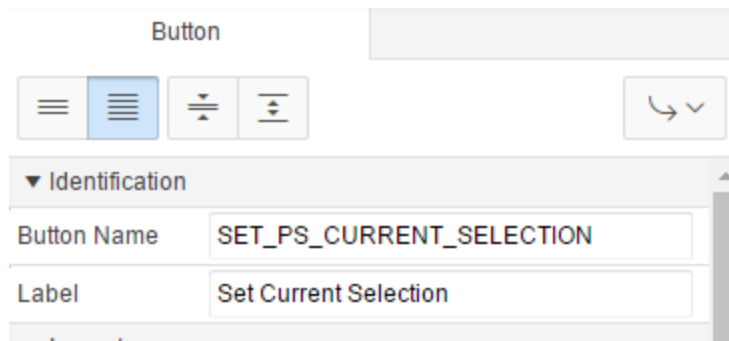

Create the "Set Current Selection" button
- Back in your new report, right-click on "Breadcrumb" and choose "Create Button"
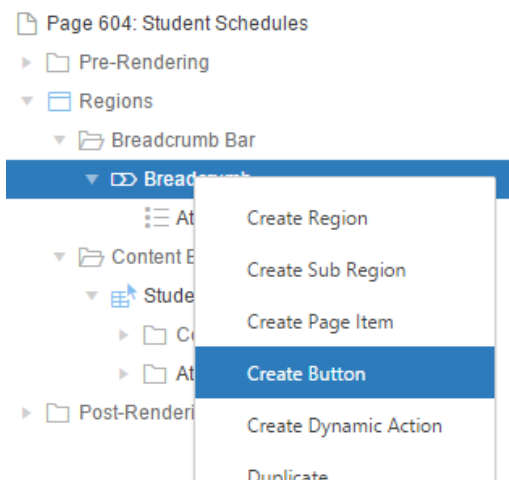
- To the right, complete the following in the "Identification" section:
  - **Button Name**: SET_PS_CURRENT_SELECTION
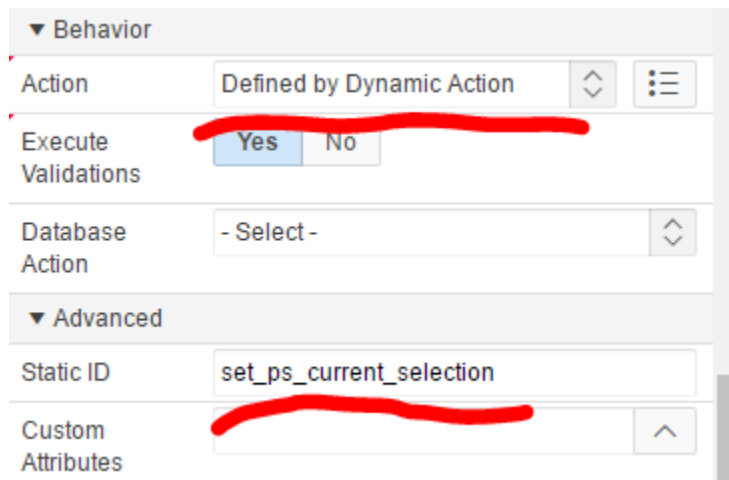 - **Label**: Set Current Selection

- 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

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