Run Query

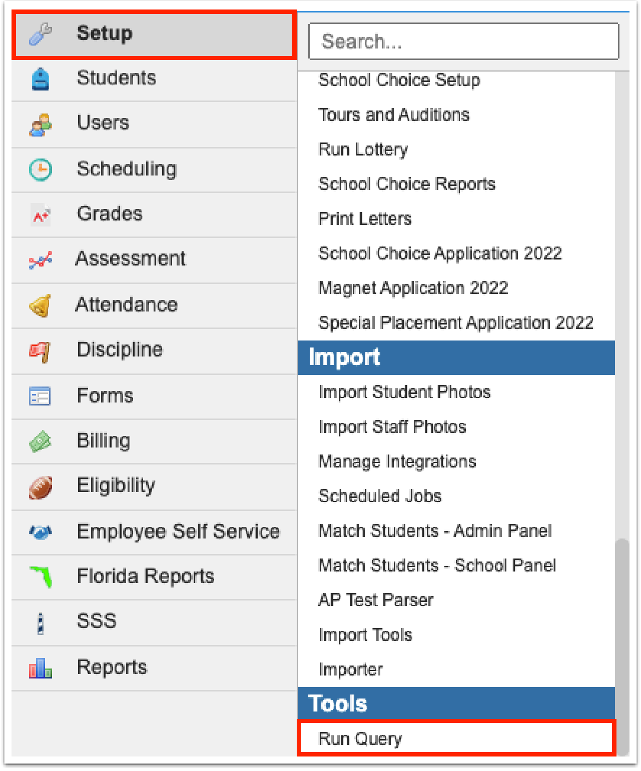
The Run Query tool is used to run custom made queries in order to review student data, user data, scheduling data, and more. The Run Query tool can also be used to update, insert, or delete date depending on profile permissions.

ℹ In order to access Run Query, the district must enable permissions either via the configuration file or via Users > Profiles > Setup tab. The commands you can use also depend on the profile permissions enabled here; options include: Select, Explain, Set, Insert, Update, Delete, Truncate, Drop, Create, Alter, Begin, Commit, and Rollback.

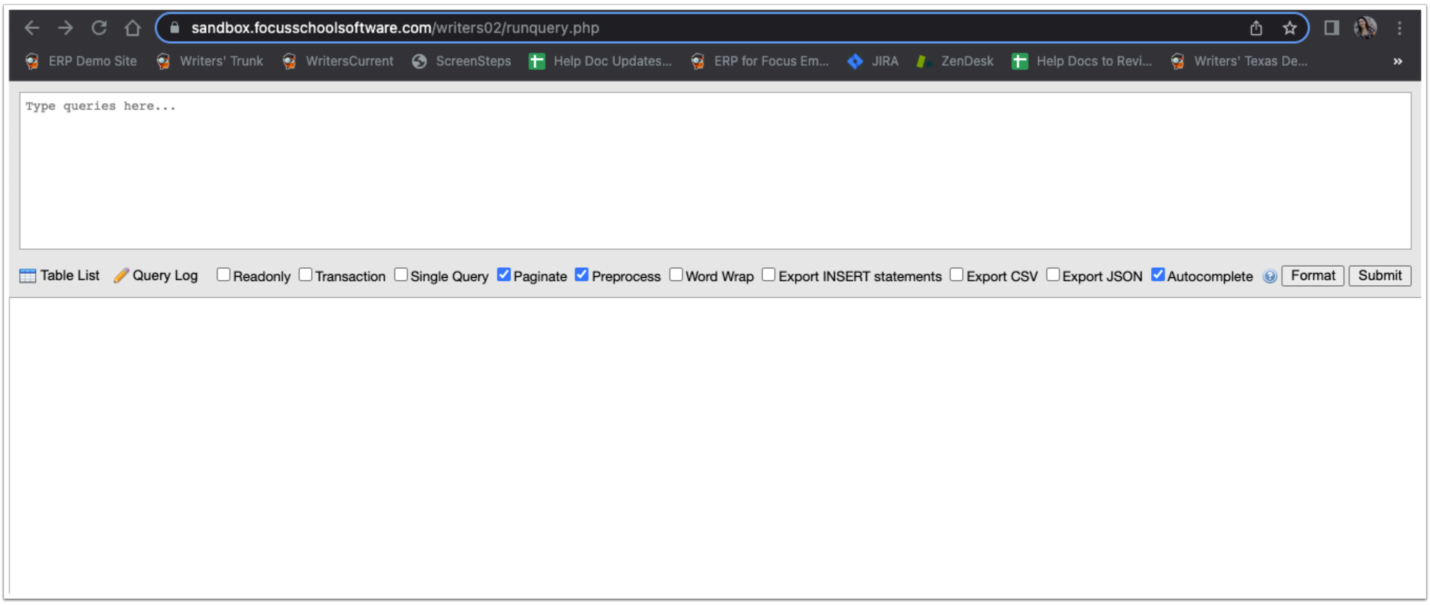
ℹ For information on obtaining your Level 1 SQL certification and on writing queries, talk to you district's Project Coordinator or Account Executive.

Run Query

1. In the **Setup** menu, click **Run Query**.

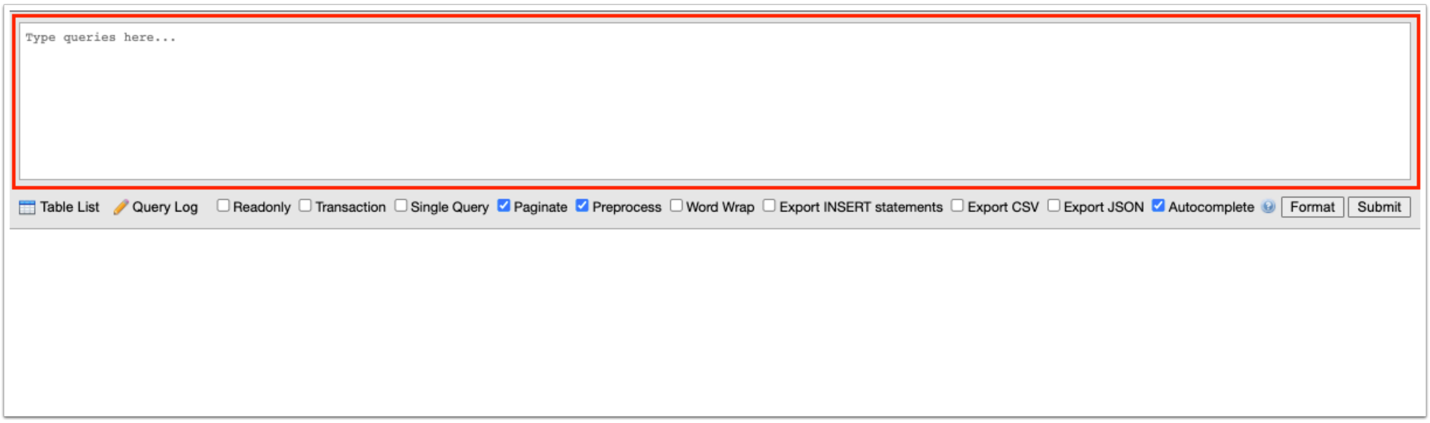


The screen reroutes to the Run Query screen, as shown in the image below.



□ Readonly □ Transaction □ Single Query ☑ Paginate ☑ Preprocess □ Word Wrap □ Export INSERT statements □ Export CSV □ Export JSON ☑ Autocomplete

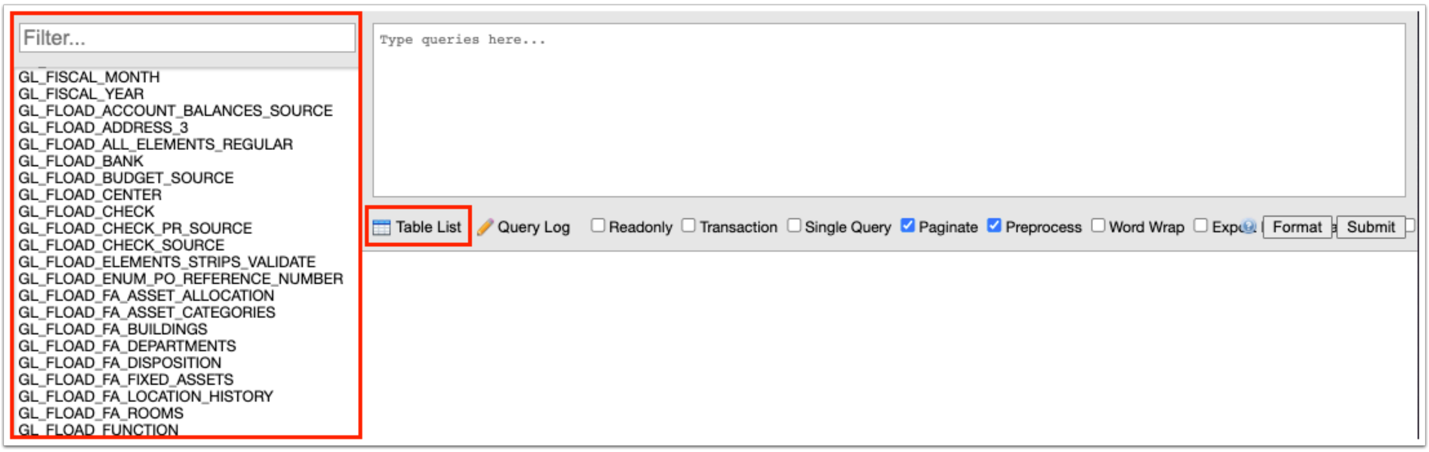
2. Enter queries in the text box at the top of the screen.



```
Type queries here...
```

Table List Query Log □ Readonly □ Transaction □ Single Query ☑ Paginate ☑ Preprocess □ Word Wrap □ Export INSERT statements □ Export CSV □ Export JSON ☑ Autocomplete Format Submit

3. To aid in writing queries, click the **Table List** for a complete list of tables.



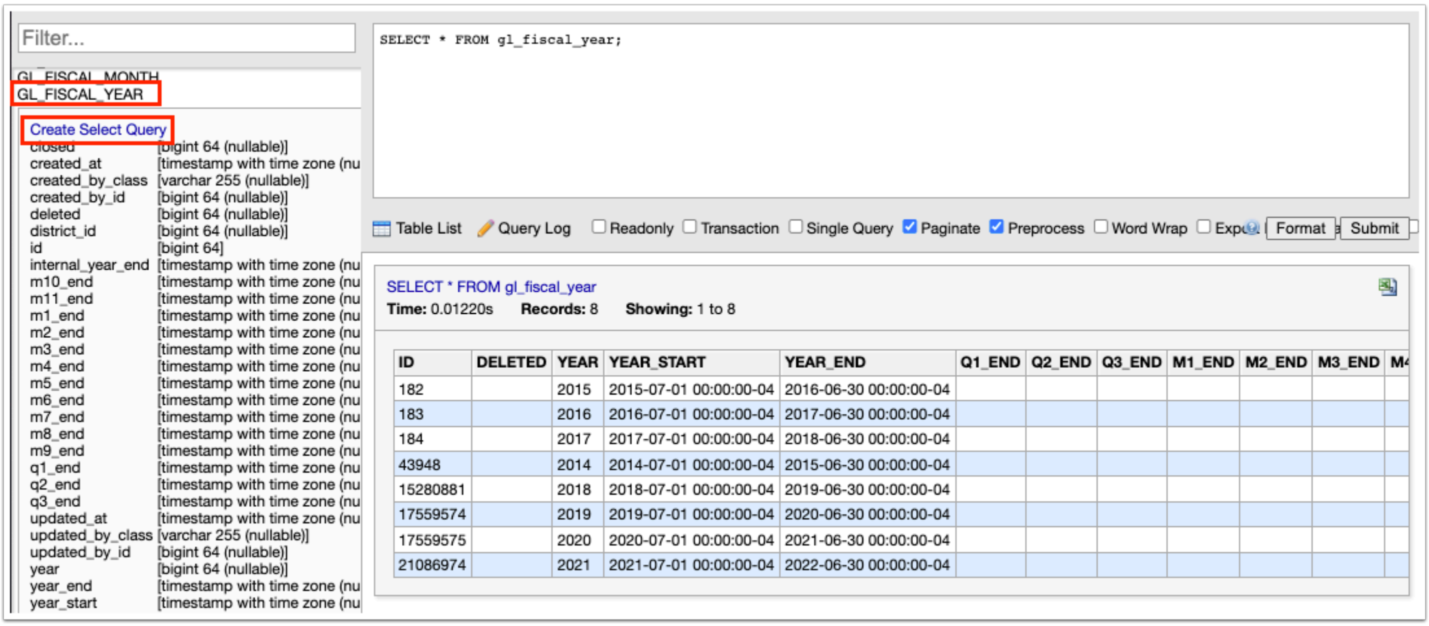
Filter...

GL_FISCAL_MONTH
GL_FISCAL_YEAR
GL_FLOAD_ACCOUNT_BALANCES_SOURCE
GL_FLOAD_ADDRESS_3
GL_FLOAD_ALL_ELEMENTS_REGULAR
GL_FLOAD_BANK
GL_FLOAD_BUDGET_SOURCE
GL_FLOAD_CENTER
GL_FLOAD_CHECK
GL_FLOAD_CHECK_PR_SOURCE
GL_FLOAD_CHECK_SOURCE
GL_FLOAD_ELEMENTS_STRIPS_VALIDATE
GL_FLOAD_ENUM_PO_REFERENCE_NUMBER
GL_FLOAD_FA_ASSET_ALLOCATION
GL_FLOAD_FA_ASSET_CATEGORIES
GL_FLOAD_FA_BUILDINGS
GL_FLOAD_FA_DEPARTMENTS
GL_FLOAD_FA_DISPOSITION
GL_FLOAD_FA_FIXED_ASSETS
GL_FLOAD_FA_LOCATION_HISTORY
GL_FLOAD_FA_ROOMS
GL_FLOAD_FUNCTION

```
Type queries here...
```

Table List Query Log □ Readonly □ Transaction □ Single Query ☑ Paginate ☑ Preprocess □ Word Wrap □ Expo Format Submit

4. Click the listed table of interest to open associated fields. You can then click the **Create Select Query** to automatically generate a query to pull data as it pertains to the select table.



Filter...

GL_FISCAL_MONTH
GL_FISCAL_YEAR

Create Select Query

closed	[bigint 64 (nullable)]
created_at	[timestamp with time zone (nu
created_by_class	[varchar 255 (nullable)]
created_by_id	[bigint 64 (nullable)]
deleted	[bigint 64 (nullable)]
district_id	[bigint 64 (nullable)]
id	[bigint 64]
internal_year_end	[timestamp with time zone (nu
m10_end	[timestamp with time zone (nu
m11_end	[timestamp with time zone (nu
m1_end	[timestamp with time zone (nu
m2_end	[timestamp with time zone (nu
m3_end	[timestamp with time zone (nu
m4_end	[timestamp with time zone (nu
m5_end	[timestamp with time zone (nu
m6_end	[timestamp with time zone (nu
m7_end	[timestamp with time zone (nu
m8_end	[timestamp with time zone (nu
m9_end	[timestamp with time zone (nu
q1_end	[timestamp with time zone (nu
q2_end	[timestamp with time zone (nu
q3_end	[timestamp with time zone (nu
updated_at	[timestamp with time zone (nu
updated_by_class	[varchar 255 (nullable)]
updated_by_id	[bigint 64 (nullable)]
year	[bigint 64 (nullable)]
year_end	[timestamp with time zone (nu
year_start	[timestamp with time zone (nu

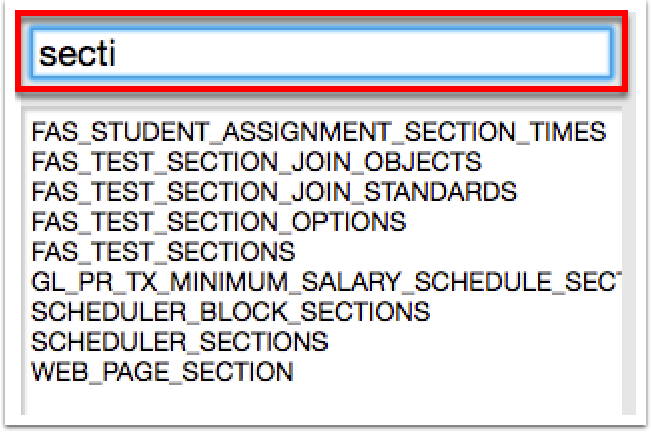
```
SELECT * FROM gl_fiscal_year;
```

Table List Query Log □ Readonly □ Transaction □ Single Query ☑ Paginate ☑ Preprocess □ Word Wrap □ Expo Format Submit

SELECT * FROM gl_fiscal_year
Time: 0.01220s **Records:** 8 **Showing:** 1 to 8

ID	DELETED	YEAR	YEAR_START	YEAR_END	Q1_END	Q2_END	Q3_END	M1_END	M2_END	M3_END	M4
182		2015	2015-07-01 00:00:00-04	2016-06-30 00:00:00-04							
183		2016	2016-07-01 00:00:00-04	2017-06-30 00:00:00-04							
184		2017	2017-07-01 00:00:00-04	2018-06-30 00:00:00-04							
43948		2014	2014-07-01 00:00:00-04	2015-06-30 00:00:00-04							
15280881		2018	2018-07-01 00:00:00-04	2019-06-30 00:00:00-04							
17559574		2019	2019-07-01 00:00:00-04	2020-06-30 00:00:00-04							
17559575		2020	2020-07-01 00:00:00-04	2021-06-30 00:00:00-04							
21086974		2021	2021-07-01 00:00:00-04	2022-06-30 00:00:00-04							

Start typing the name of a table or a field to find the corresponding table in the **Filter** text box.

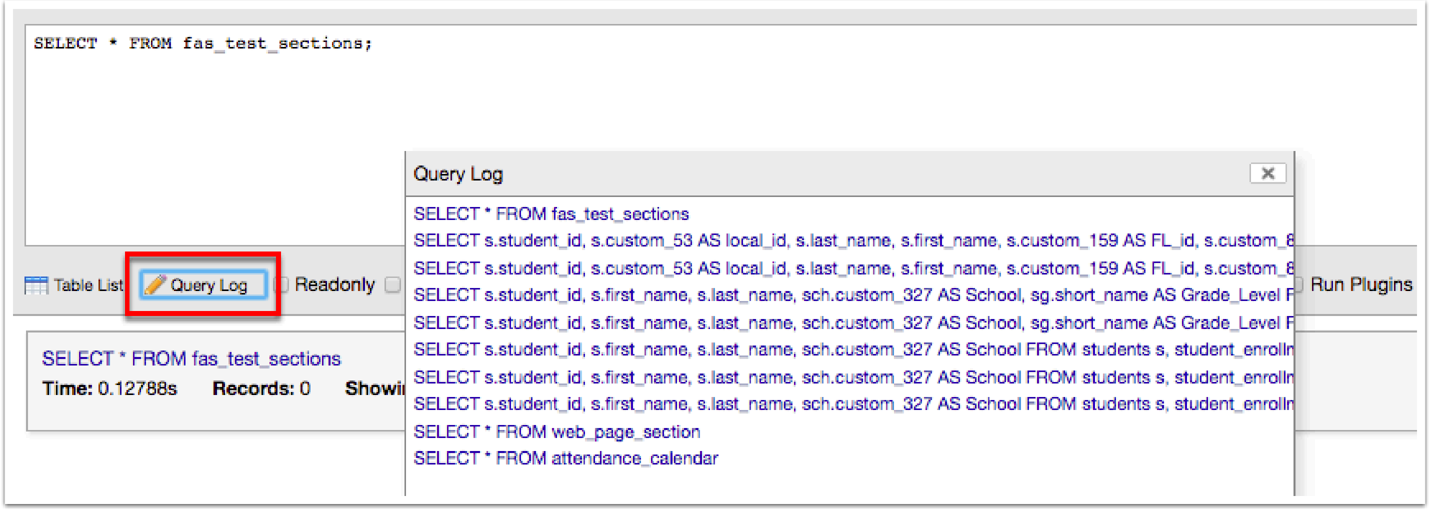


secti

```
FAS_STUDENT_ASSIGNMENT_SECTION_TIMES
FAS_TEST_SECTION_JOIN_OBJECTS
FAS_TEST_SECTION_JOIN_STANDARDS
FAS_TEST_SECTION_OPTIONS
FAS_TEST_SECTIONS
GL_PR_TX_MINIMUM_SALARY_SCHEDULE_SEC
SCHEDULER_BLOCK_SECTIONS
SCHEDULER_SECTIONS
WEB_PAGE_SECTION
```

To close the **Table List** click the icon again.

5. Click the **Query Log** icon to open a history of the last 25 queries ran.



```
SELECT * FROM fas_test_sections;
```

Query Log ✕

```
SELECT * FROM fas_test_sections
SELECT s.student_id, s.custom_53 AS local_id, s.last_name, s.first_name, s.custom_159 AS FL_id, s.custom_8
SELECT s.student_id, s.custom_53 AS local_id, s.last_name, s.first_name, s.custom_159 AS FL_id, s.custom_8
SELECT s.student_id, s.first_name, s.last_name, sch.custom_327 AS School, sg.short_name AS Grade_Level F
SELECT s.student_id, s.first_name, s.last_name, sch.custom_327 AS School, sg.short_name AS Grade_Level F
SELECT s.student_id, s.first_name, s.last_name, sch.custom_327 AS School FROM students s, student_enrolln
SELECT s.student_id, s.first_name, s.last_name, sch.custom_327 AS School FROM students s, student_enrolln
SELECT s.student_id, s.first_name, s.last_name, sch.custom_327 AS School FROM students s, student_enrolln
SELECT * FROM web_page_section
SELECT * FROM attendance_calendar
```

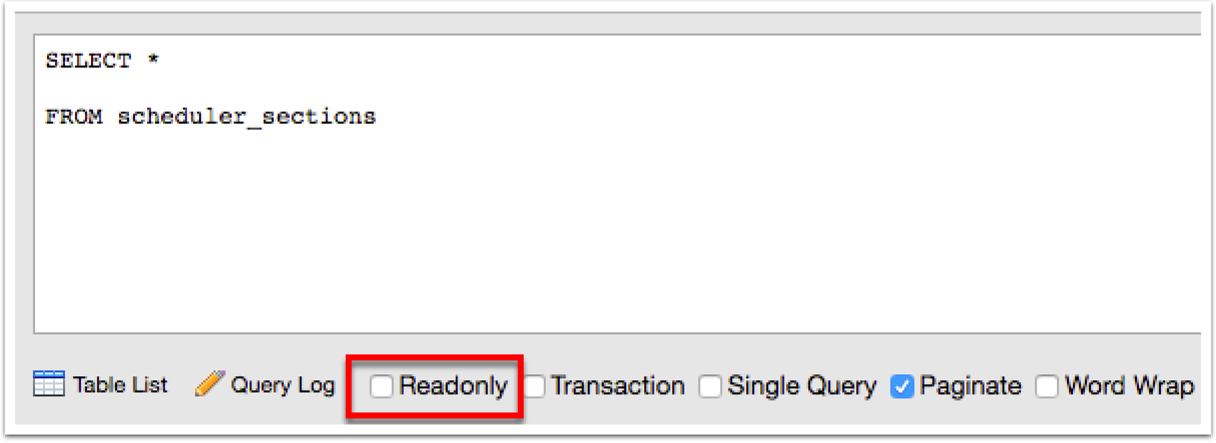
SELECT * FROM fas_test_sections

Time: 0.12788s **Records:** 0 **Showi**

To open the query from your log, click the query link, which will populate the query text box automatically.

Click the **X** to close the Query Log.

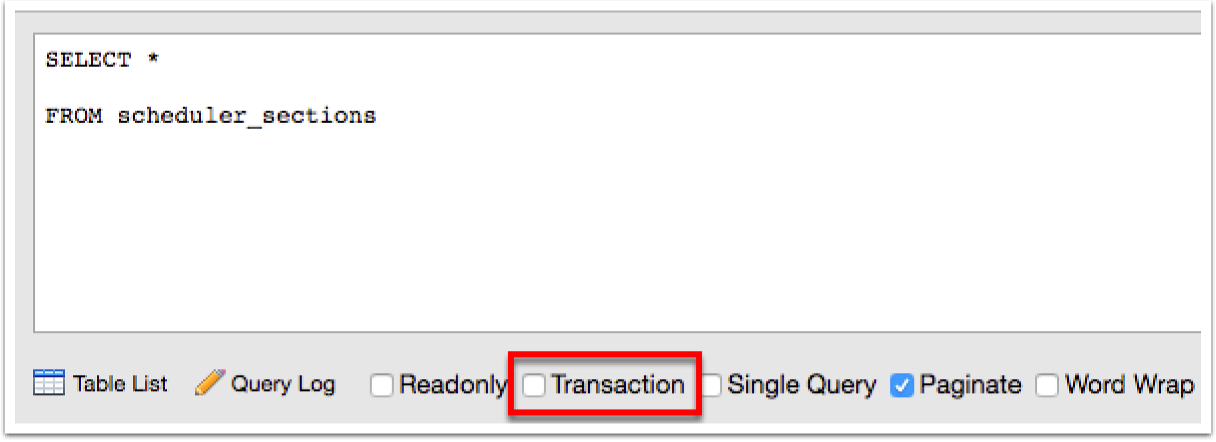
6. Click the **Readonly** text box to run queries without making actual changes to the database, which is useful when running queries that will be used to update, insert, or delete data.



```
SELECT *

FROM scheduler_sections
```

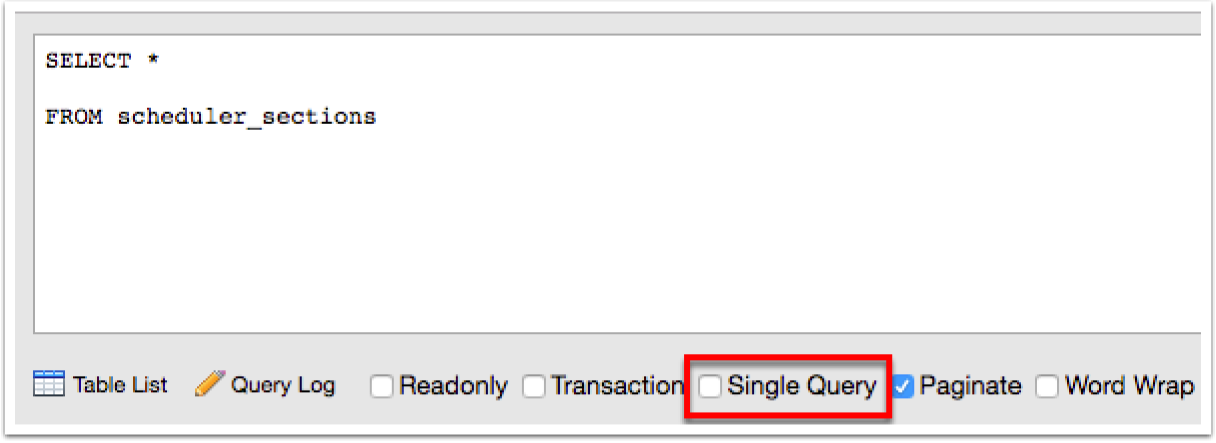
7. Click the **Transaction** check box to make all of the queries you've typed in the run text box into a single transaction instead of having a separate transactions for each query. This is useful if you're testing something and need to be able to use the ROLLBACK, TRY, etc. functionality of SQL. I.e. The essential point of a transaction is that it bundles multiple steps into a single, all-or-nothing operation. The intermediate states between the steps are not visible to other concurrent transactions, and if some failure occurs that prevents the transaction from completing, then none of the steps affect the database at all.



```
SELECT *

FROM scheduler_sections
```

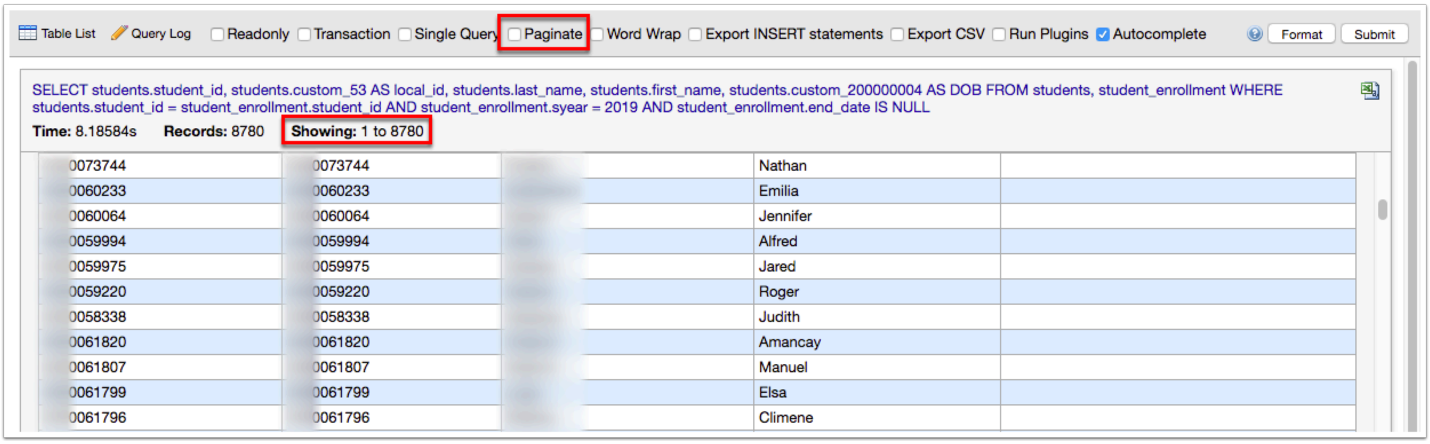
8. Click the **Single Query** check box to take every line of code and run it all at once as if it were a single query. This is important when you're creating things like stored procedures or functions since they can, and often do, use more than one query. If the Single Query check box is not selected, the PDO driver (the protocol that transmits the code you type to the database) won't know that the code should be run as a single query and will instead try and run every individual query inside it.



```
SELECT *

FROM scheduler_sections
```

9. Select the **Paginate** check box to break down query results into pages as opposed to having all results displays on a single page where you would have to scroll through all of the results. The example displayed below shows results when not paginated.

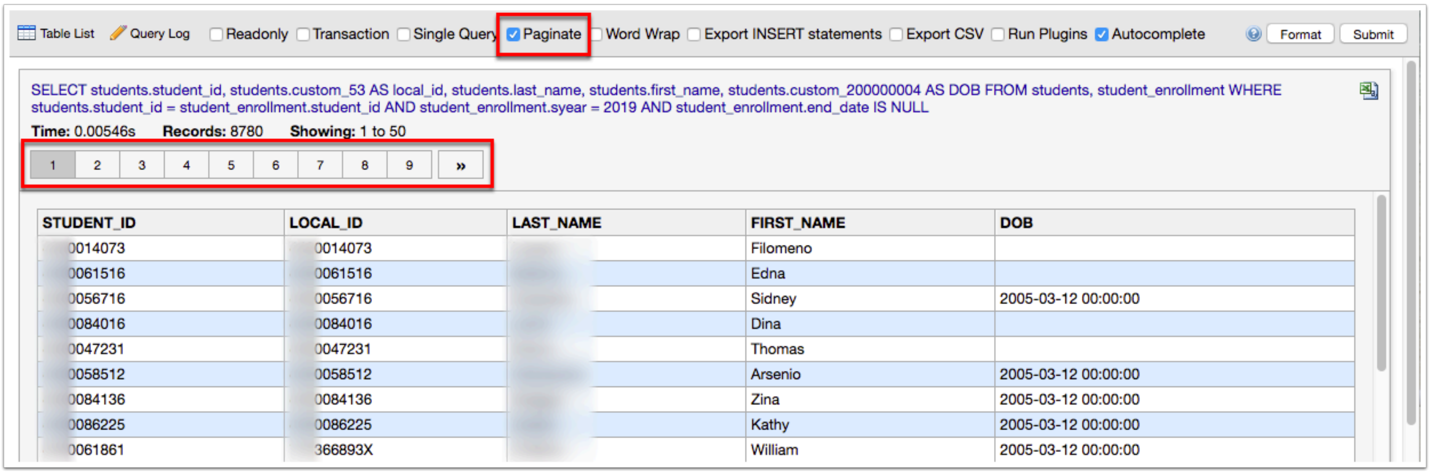


```
SELECT students.student_id, students.custom_53 AS local_id, students.last_name, students.first_name, students.custom_200000004 AS DOB FROM students, student_enrollment WHERE students.student_id = student_enrollment.student_id AND student_enrollment.syear = 2019 AND student_enrollment.end_date IS NULL
```

Time: 8.18584s **Records:** 8780 **Showing:** 1 to 8780

0073744	0073744	[redacted]	Nathan	
0060233	0060233	[redacted]	Emilia	
0060064	0060064	[redacted]	Jennifer	
0059994	0059994	[redacted]	Alfred	
0059975	0059975	[redacted]	Jared	
0059220	0059220	[redacted]	Roger	
0058338	0058338	[redacted]	Judith	
0061820	0061820	[redacted]	Amancay	
0061807	0061807	[redacted]	Manuel	
0061799	0061799	[redacted]	Elsa	
0061796	0061796	[redacted]	Climene	

The image below shows results when the Paginate check box is selected.

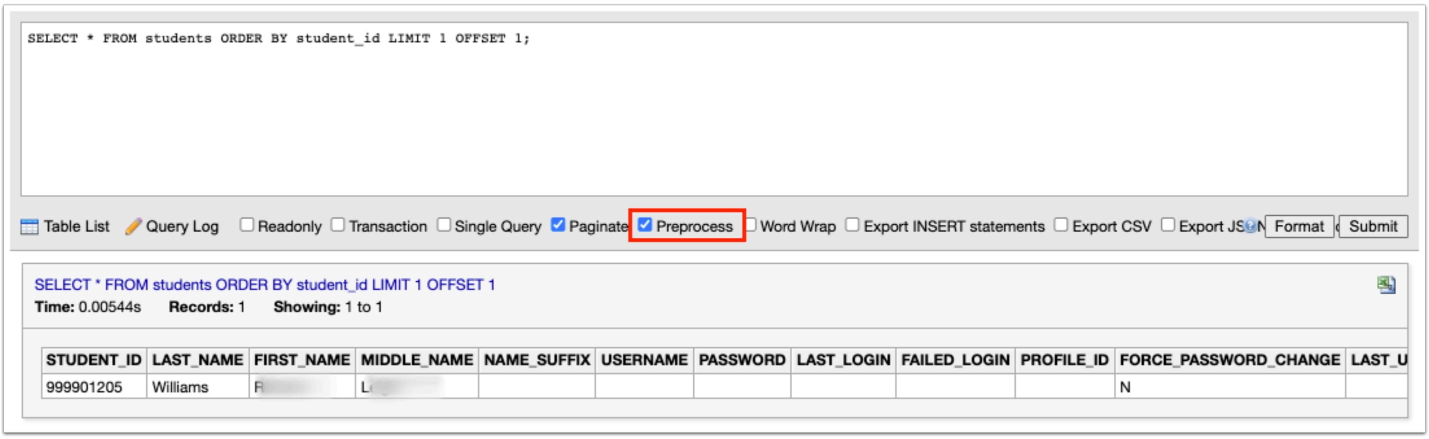


```
SELECT students.student_id, students.custom_53 AS local_id, students.last_name, students.first_name, students.custom_200000004 AS DOB FROM students, student_enrollment WHERE students.student_id = student_enrollment.student_id AND student_enrollment.syear = 2019 AND student_enrollment.end_date IS NULL
```

Time: 0.00546s **Records:** 8780 **Showing:** 1 to 50

STUDENT_ID	LOCAL_ID	LAST_NAME	FIRST_NAME	DOB
0014073	0014073	[redacted]	Filomeno	
0061516	0061516	[redacted]	Edna	
0056716	0056716	[redacted]	Sidney	2005-03-12 00:00:00
0084016	0084016	[redacted]	Dina	
0047231	0047231	[redacted]	Thomas	
0058512	0058512	[redacted]	Arsenio	2005-03-12 00:00:00
0084136	0084136	[redacted]	Zina	2005-03-12 00:00:00
0086225	0086225	[redacted]	Kathy	2005-03-12 00:00:00
0061861	366893X	[redacted]	William	2005-03-12 00:00:00

10. Select the **Preprocess** check box to run Postgres type casts in the SQL Server. Empty arrays will be converted to NULL instead of throwing a SQL syntax error. Queries with LIMIT/OFFSET are converted to the cross-DB OFFSET/FETCH syntax automatically.



```
SELECT * FROM students ORDER BY student_id LIMIT 1 OFFSET 1;
```

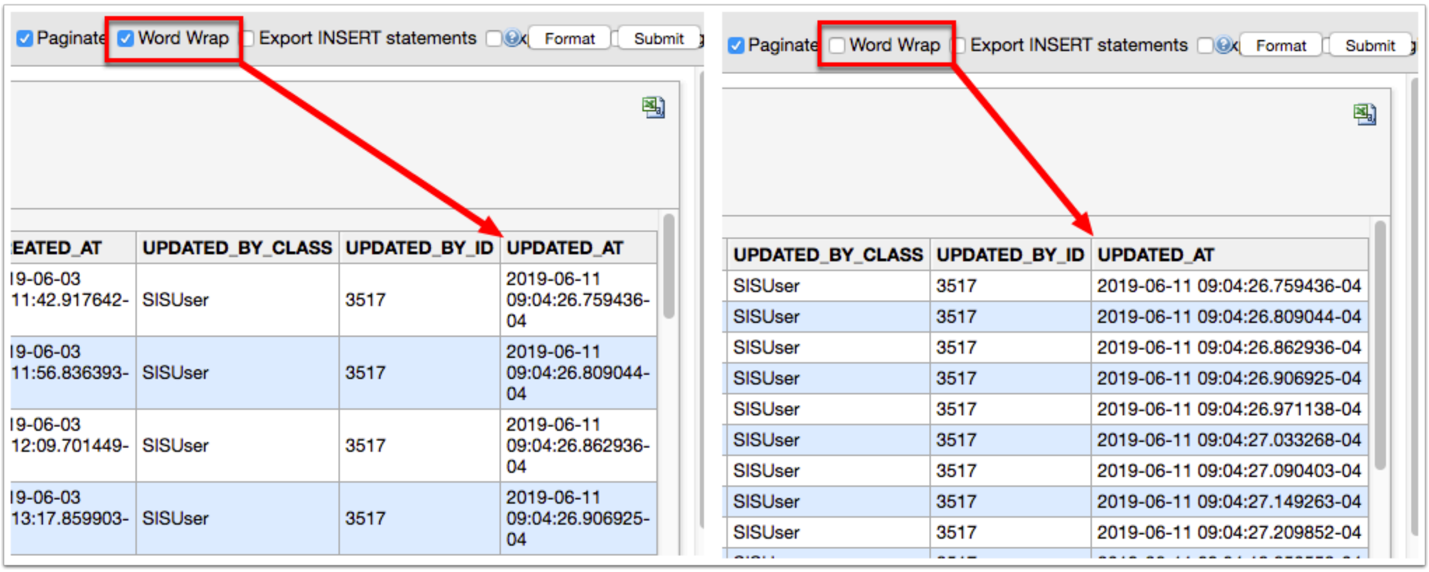
Table List · Query Log · □ Readonly □ Transaction □ Single Query ☑ Paginate ☑ Preprocess □ Word Wrap □ Export INSERT statements □ Export CSV □ Export JSON · Format · Submit

SELECT * FROM students ORDER BY student_id LIMIT 1 OFFSET 1

Time: 0.00544s **Records:** 1 **Showing:** 1 to 1

STUDENT_ID	LAST_NAME	FIRST_NAME	MIDDLE_NAME	NAME_SUFFIX	USERNAME	PASSWORD	LAST_LOGIN	FAILED_LOGIN	PROFILE_ID	FORCE_PASSWORD_CHANGE	LAST_U
999901205	Williams	F[redacted]	L[redacted]							N	

11. Select the **Word Wrap** check box to format the text in the query results in word wrap; i.e. if the information goes beyond the provided area, you can wrap the text and extend the box to make the information more readable.



☑ Paginate ☑ Word Wrap □ Export INSERT statements □ · Format · Submit

EATED_AT	UPDATED_BY_CLASS	UPDATED_BY_ID	UPDATED_AT
19-06-03 11:42.917642-	SISUser	3517	2019-06-11 09:04:26.759436-04
19-06-03 11:56.836393-	SISUser	3517	2019-06-11 09:04:26.809044-04
19-06-03 12:09.701449-	SISUser	3517	2019-06-11 09:04:26.862936-04
19-06-03 13:17.859903-	SISUser	3517	2019-06-11 09:04:26.906925-04

☑ Paginate □ Word Wrap □ Export INSERT statements □ · Format · Submit

UPDATED_BY_CLASS	UPDATED_BY_ID	UPDATED_AT
SISUser	3517	2019-06-11 09:04:26.759436-04
SISUser	3517	2019-06-11 09:04:26.809044-04
SISUser	3517	2019-06-11 09:04:26.862936-04
SISUser	3517	2019-06-11 09:04:26.906925-04
SISUser	3517	2019-06-11 09:04:26.971138-04
SISUser	3517	2019-06-11 09:04:27.033268-04
SISUser	3517	2019-06-11 09:04:27.090403-04
SISUser	3517	2019-06-11 09:04:27.149263-04
SISUser	3517	2019-06-11 09:04:27.209852-04

12. Select the **Export INSERT statements** check box to format the output of a SELECT query into an INSERT query syntax with a [table] placeholder, which allows you to update data using the table to which data is to be inserted.

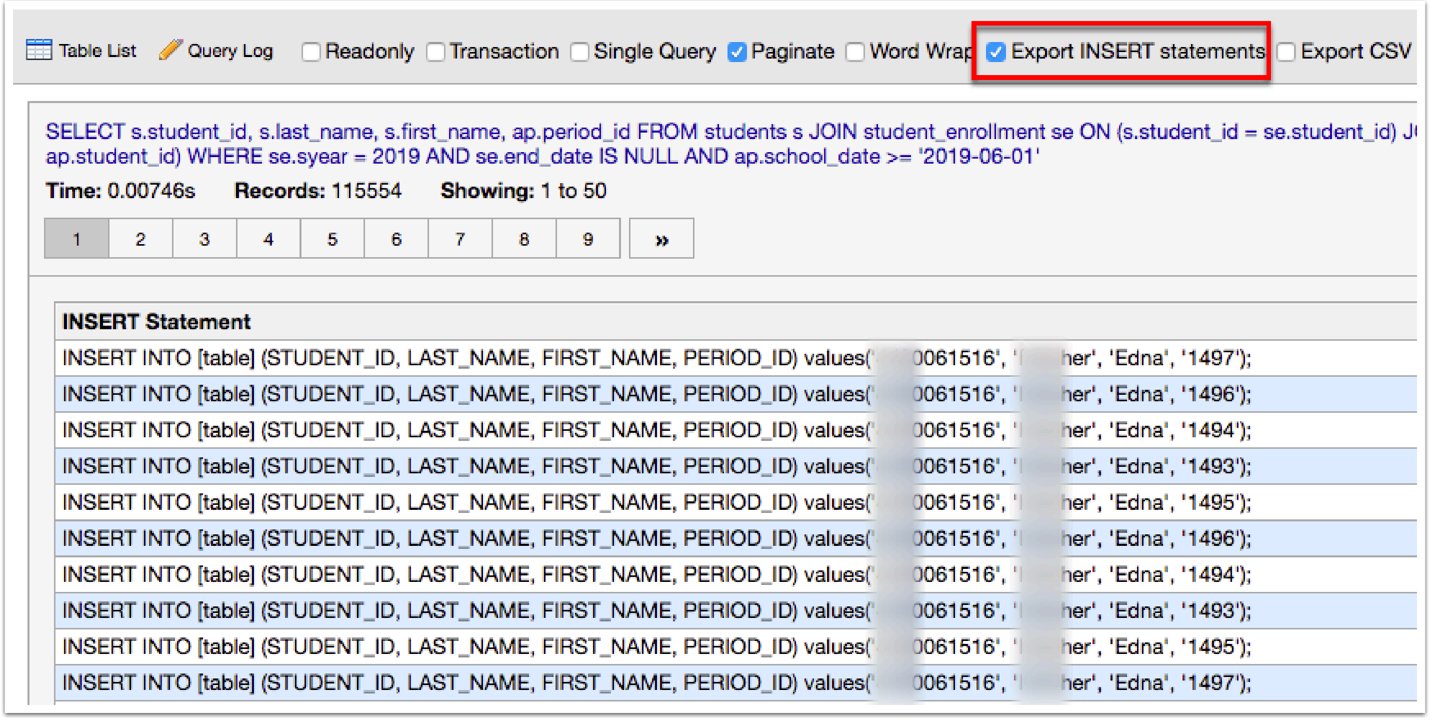


Table List · Query Log · □ Readonly □ Transaction □ Single Query ☑ Paginate □ Word Wrap ☑ Export INSERT statements □ Export CSV

SELECT s.student_id, s.last_name, s.first_name, ap.period_id FROM students s JOIN student_enrollment se ON (s.student_id = se.student_id) J
ap.student_id) WHERE se.syear = 2019 AND se.end_date IS NULL AND ap.school_date >= '2019-06-01'

Time: 0.00746s **Records:** 115554 **Showing:** 1 to 50

1 2 3 4 5 6 7 8 9 »

INSERT Statement
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1497');
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1496');
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1494');
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1493');
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1495');
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1496');
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1494');
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1493');
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1495');
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1497');

13. Select the **Export CSV** check box to download the query results in a CSV file to your computer.

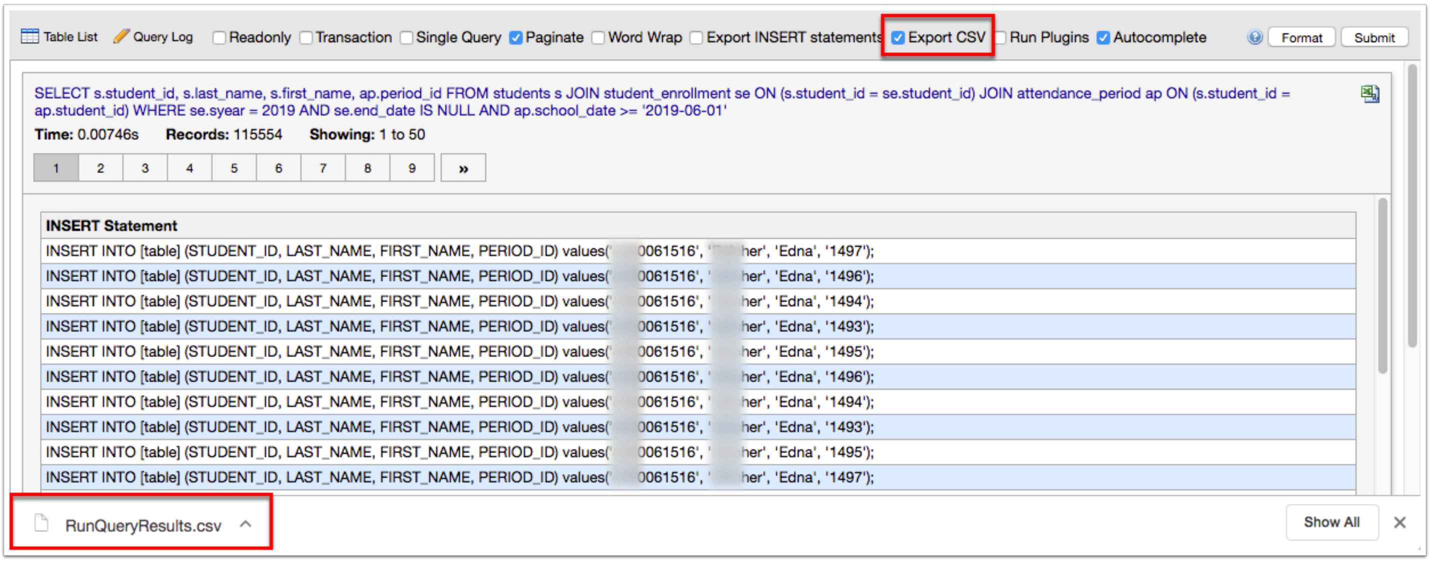


Table List · Query Log · □ Readonly □ Transaction □ Single Query ☑ Paginate □ Word Wrap □ Export INSERT statements ☑ Export CSV □ Run Plugins ☑ Autocomplete · Format · Submit

SELECT s.student_id, s.last_name, s.first_name, ap.period_id FROM students s JOIN student_enrollment se ON (s.student_id = se.student_id) JOIN attendance_period ap ON (s.student_id = ap.student_id) WHERE se.syear = 2019 AND se.end_date IS NULL AND ap.school_date >= '2019-06-01'

Time: 0.00746s **Records:** 115554 **Showing:** 1 to 50

1 2 3 4 5 6 7 8 9 »

INSERT Statement
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1497');
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1496');
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1494');
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1493');
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1495');
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1496');
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1494');
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1493');
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1495');
INSERT INTO [table] (STUDENT_ID, LAST_NAME, FIRST_NAME, PERIOD_ID) values('[redacted]0061516', '[redacted]her', 'Edna', '1497');

RunQueryResults.csv · Show All ×

14. Select the **Export JSON** check box to download the query results in a JSON file (JavaScript Object Notation).

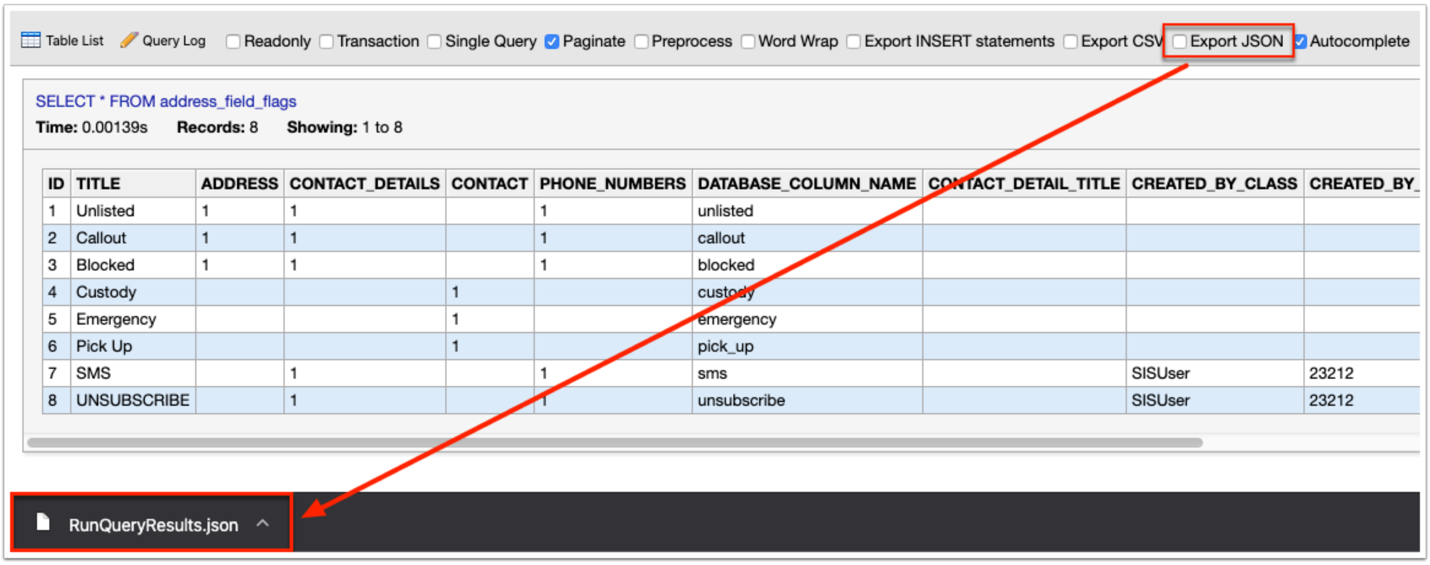


Table List · Query Log · □ Readonly □ Transaction □ Single Query ☑ Paginate □ Preprocess □ Word Wrap □ Export INSERT statements □ Export CSV □ Export JSON ☑ Autocomplete

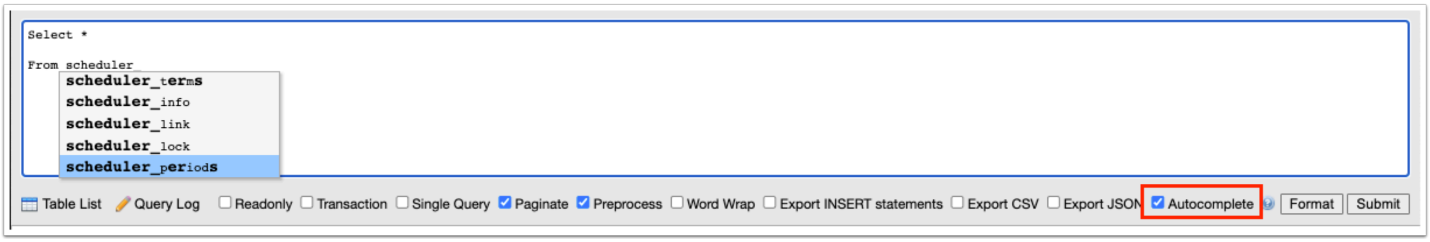
SELECT * FROM address_field_flags

Time: 0.00139s **Records:** 8 **Showing:** 1 to 8

ID	TITLE	ADDRESS	CONTACT_DETAILS	CONTACT	PHONE_NUMBERS	DATABASE_COLUMN_NAME	CONTACT_DETAIL_TITLE	CREATED_BY_CLASS	CREATED_BY_
1	Unlisted	1	1		1	unlisted			
2	Callout	1	1		1	callout			
3	Blocked	1	1		1	blocked			
4	Custody			1		custody			
5	Emergency			1		emergency			
6	Pick Up			1		pick_up			
7	SMS		1		1	sms		SISUser	23212
8	UNSUBSCRIBE		1		1	unsubscribe		SISUser	23212

RunQueryResults.json

15. Select the **Autocomplete** check box to allow the system to make recommendation upon typing queries; i.e. when you begin typing the name of a table, several possible options will displays based on the information typed. From here, you can select the applicable option instead of having to type it out.

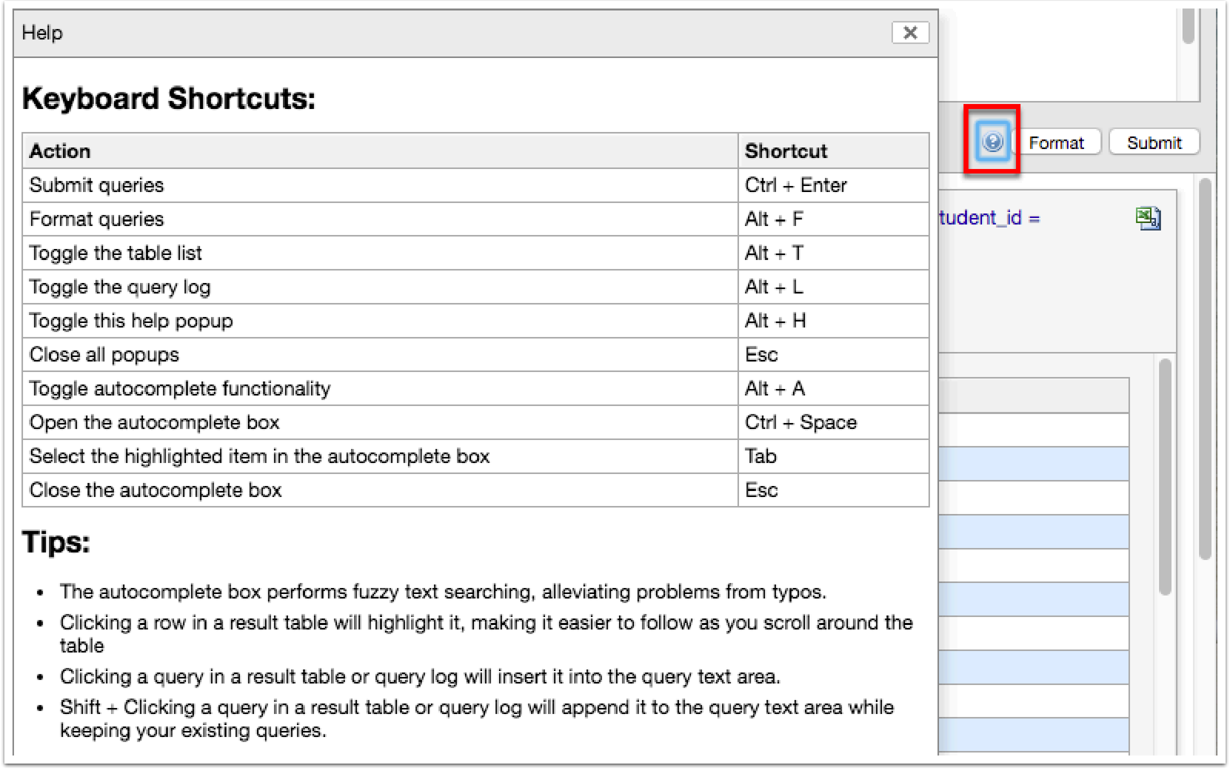


```
Select *

From scheduler
     scheduler_terms
     scheduler_info
     scheduler_link
     scheduler_lock
     scheduler_periods
```

Table List · Query Log · □ Readonly □ Transaction □ Single Query ☑ Paginate ☑ Preprocess □ Word Wrap □ Export INSERT statements □ Export CSV □ Export JSON ☑ Autocomplete · Format · Submit

16. Click the **?** icon for Help, which includes Keyboard Shortcuts and Tips.



Keyboard Shortcuts:

Action	Shortcut
Submit queries	Ctrl + Enter
Format queries	Alt + F
Toggle the table list	Alt + T
Toggle the query log	Alt + L
Toggle this help popup	Alt + H
Close all popups	Esc
Toggle autocomplete functionality	Alt + A
Open the autocomplete box	Ctrl + Space
Select the highlighted item in the autocomplete box	Tab
Close the autocomplete box	Esc

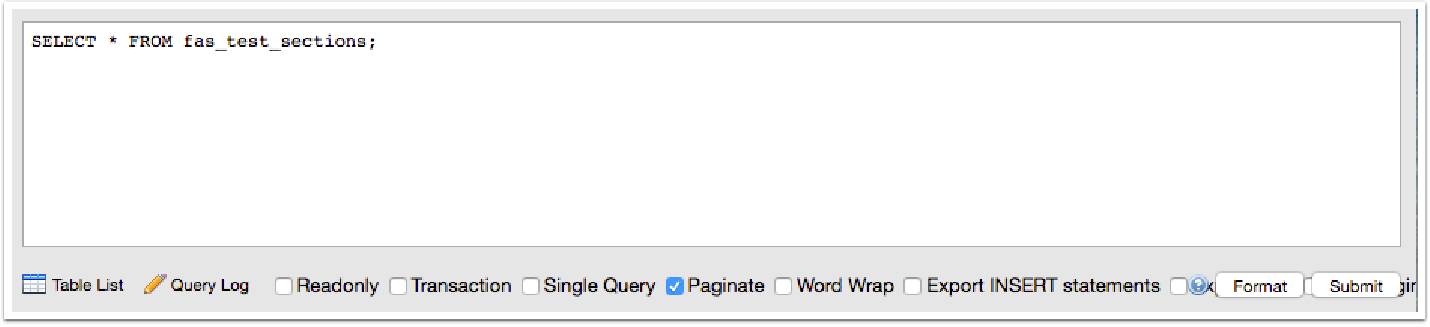
Tips:

- The autocomplete box performs fuzzy text searching, alleviating problems from typos.
- Clicking a row in a result table will highlight it, making it easier to follow as you scroll around the table
- Clicking a query in a result table or query log will insert it into the query text area.
- Shift + Clicking a query in a result table or query log will append it to the query text area while keeping your existing queries.

Click the **X** to close the Help window.

17. Click the **Format** button after entering a query in the provided text box in order to format the query.

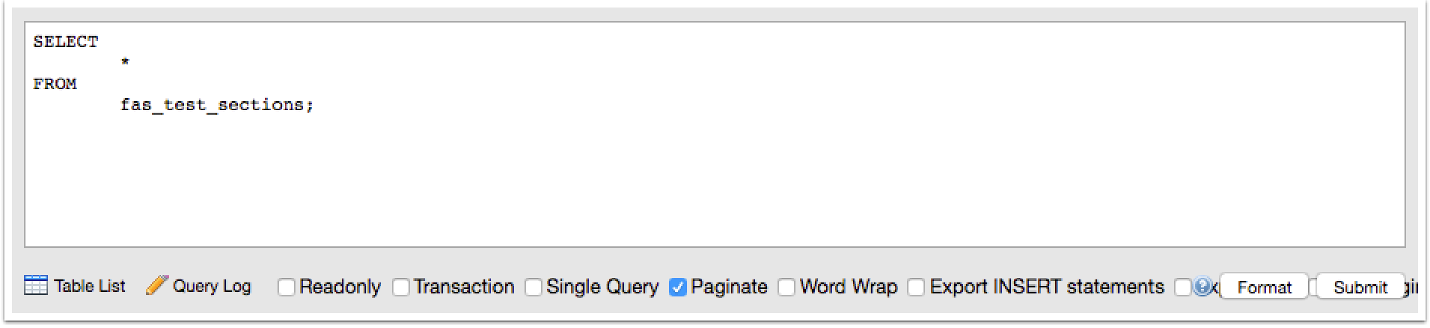
The query displayed below is an example of non-formatted.



```
SELECT * FROM fas_test_sections;
```

☐ Readonly ☐ Transaction ☐ Single Query ☑ Paginate ☐ Word Wrap ☐ Export INSERT statements ☐

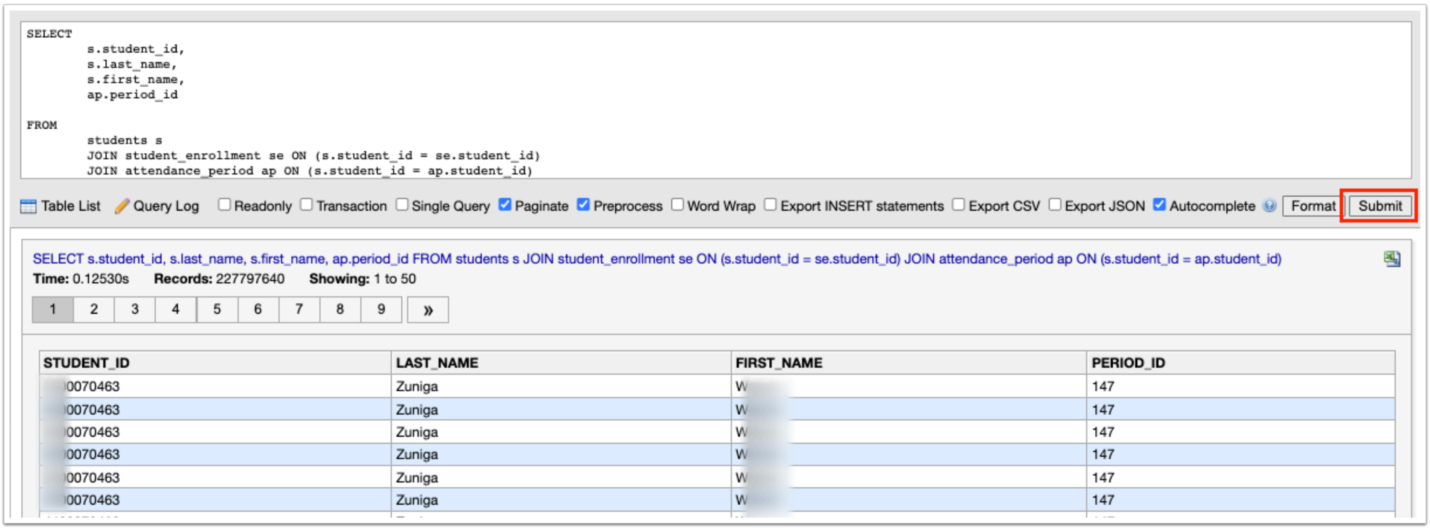
The query displayed below is an example of formatted.



```
SELECT
        *
FROM
        fas_test_sections;
```

☐ Readonly ☐ Transaction ☐ Single Query ☑ Paginate ☐ Word Wrap ☐ Export INSERT statements ☐

18. Once a query has been entered and all selections have been made, click **Submit** for query results, which display on the bottom of the screen.



```
SELECT
        s.student_id,
        s.last_name,
        s.first_name,
        ap.period_id
FROM
        students s
        JOIN student_enrollment se ON (s.student_id = se.student_id)
        JOIN attendance_period ap ON (s.student_id = ap.student_id)
```

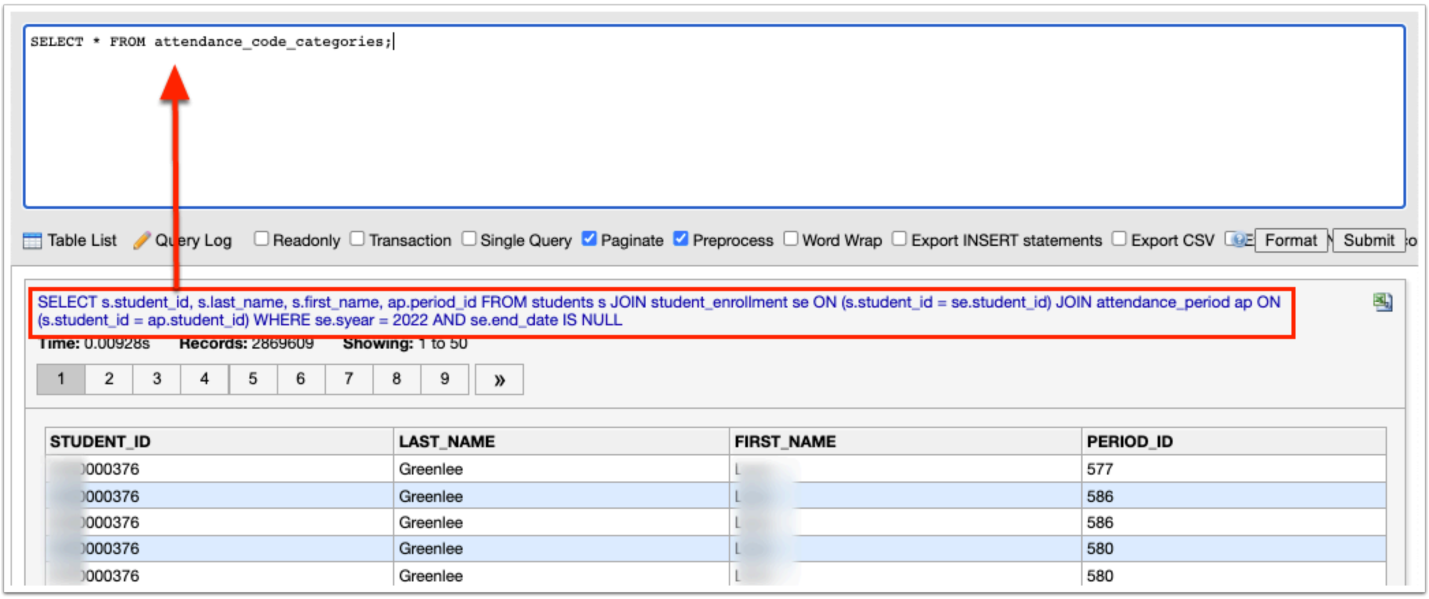
□ Readonly □ Transaction □ Single Query ☑ Paginate ☑ Preprocess □ Word Wrap □ Export INSERT statements □ Export CSV □ Export JSON ☑ Autocomplete

SELECT s.student_id, s.last_name, s.first_name, ap.period_id FROM students s JOIN student_enrollment se ON (s.student_id = se.student_id) JOIN attendance_period ap ON (s.student_id = ap.student_id)

Time: 0.12530s **Records:** 227797640 **Showing:** 1 to 50

STUDENT_ID	LAST_NAME	FIRST_NAME	PERIOD_ID
[redacted]0070463	Zuniga	W[redacted]	147
[redacted]0070463	Zuniga	W[redacted]	147
[redacted]0070463	Zuniga	W[redacted]	147
[redacted]0070463	Zuniga	W[redacted]	147
[redacted]0070463	Zuniga	W[redacted]	147
[redacted]0070463	Zuniga	W[redacted]	147

19. The results displayed is based on the query displayed. If the query is changed in the provided text box, you can repopulate the text box with the previously run query by clicking the query link.



```
SELECT * FROM attendance_code_categories;
```

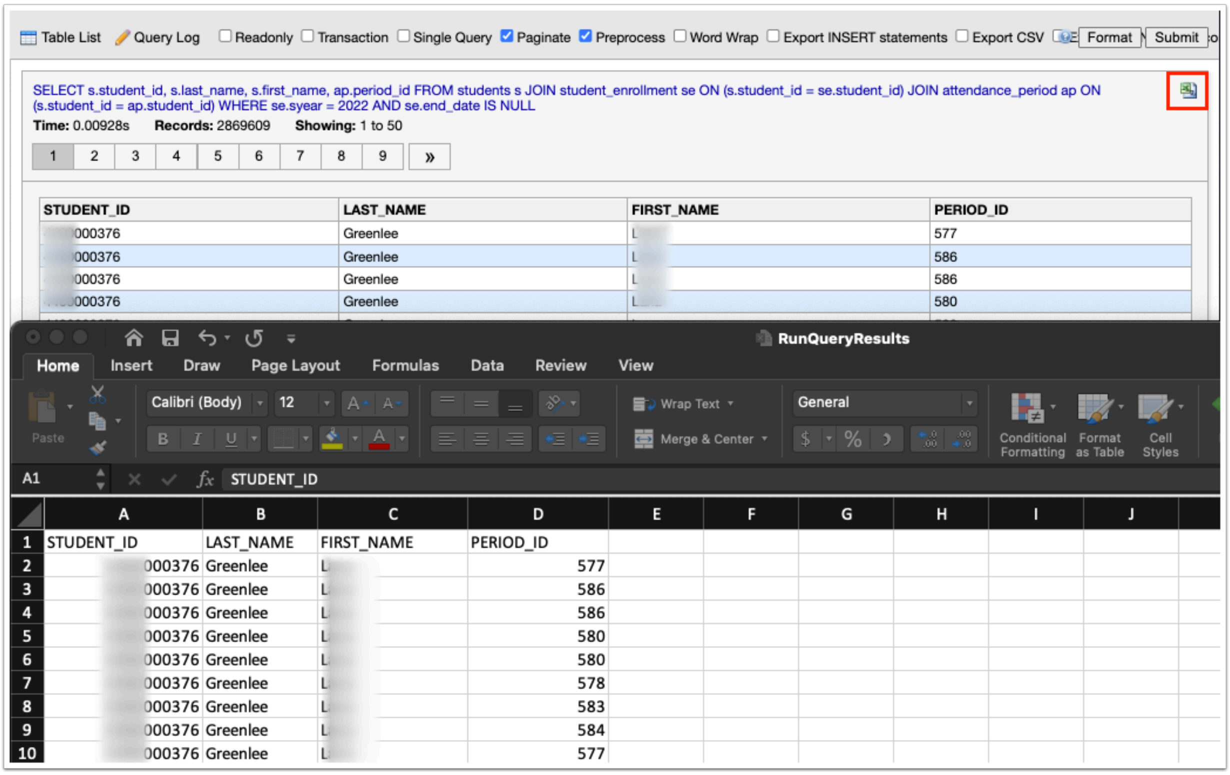
□ Readonly □ Transaction □ Single Query ☑ Paginate ☑ Preprocess □ Word Wrap □ Export INSERT statements □ Export CSV

SELECT s.student_id, s.last_name, s.first_name, ap.period_id FROM students s JOIN student_enrollment se ON (s.student_id = se.student_id) JOIN attendance_period ap ON (s.student_id = ap.student_id) WHERE se.syear = 2022 AND se.end_date IS NULL

Time: 0.00928s **Records:** 2869609 **Showing:** 1 to 50

STUDENT_ID	LAST_NAME	FIRST_NAME	PERIOD_ID
[redacted]000376	Greenlee	L[redacted]	577
[redacted]000376	Greenlee	L[redacted]	586
[redacted]000376	Greenlee	L[redacted]	586
[redacted]000376	Greenlee	L[redacted]	580
[redacted]000376	Greenlee	L[redacted]	580

20. Click the **Excel** icon to download the query results to an Excel spreadsheet.



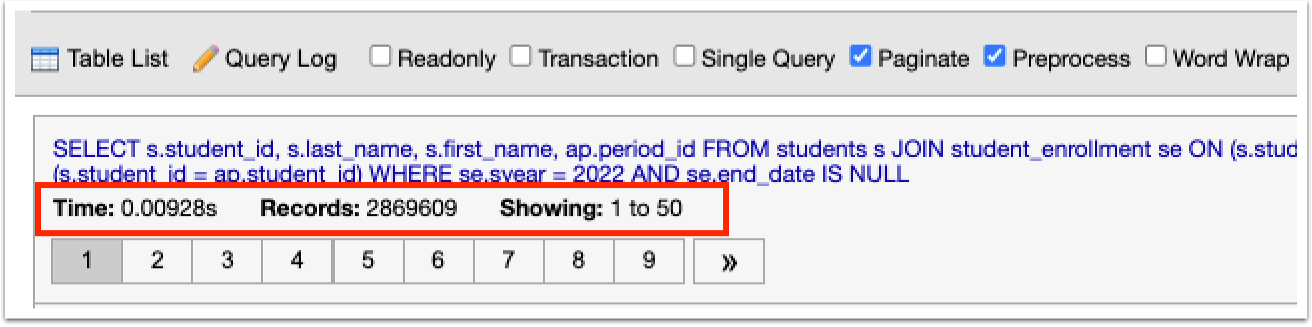
SELECT s.student_id, s.last_name, s.first_name, ap.period_id FROM students s JOIN student_enrollment se ON (s.student_id = se.student_id) JOIN attendance_period ap ON (s.student_id = ap.student_id) WHERE se.syear = 2022 AND se.end_date IS NULL

Time: 0.00928s **Records:** 2869609 **Showing:** 1 to 50

STUDENT_ID	LAST_NAME	FIRST_NAME	PERIOD_ID
[redacted]000376	Greenlee	L[redacted]	577
[redacted]000376	Greenlee	L[redacted]	586
[redacted]000376	Greenlee	L[redacted]	586
[redacted]000376	Greenlee	L[redacted]	580

	A	B	C	D	E	F	G	H	I	J
1	STUDENT_ID	LAST_NAME	FIRST_NAME	PERIOD_ID						
2	[redacted]000376	Greenlee	L[redacted]	577						
3	[redacted]000376	Greenlee	L[redacted]	586						
4	[redacted]000376	Greenlee	L[redacted]	586						
5	[redacted]000376	Greenlee	L[redacted]	580						
6	[redacted]000376	Greenlee	L[redacted]	580						
7	[redacted]000376	Greenlee	L[redacted]	578						
8	[redacted]000376	Greenlee	L[redacted]	583						
9	[redacted]000376	Greenlee	L[redacted]	584						
10	[redacted]000376	Greenlee	L[redacted]	577						

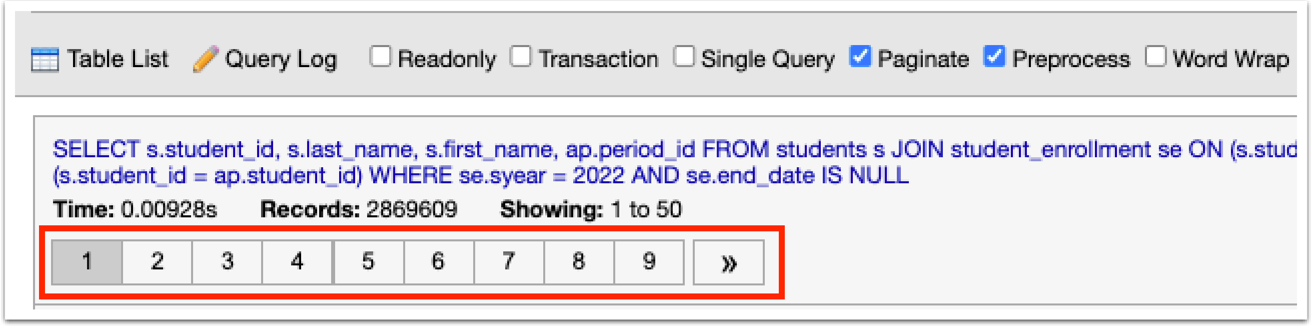
The **Time** is took for the query to pull the applicable results displays, as well as the number of **Records**, and the number of records **Showing**.



SELECT s.student_id, s.last_name, s.first_name, ap.period_id FROM students s JOIN student_enrollment se ON (s.stud
(s.student_id = ap.student_id) WHERE se.syear = 2022 AND se.end_date IS NULL

Time: 0.00928s **Records:** 2869609 **Showing:** 1 to 50

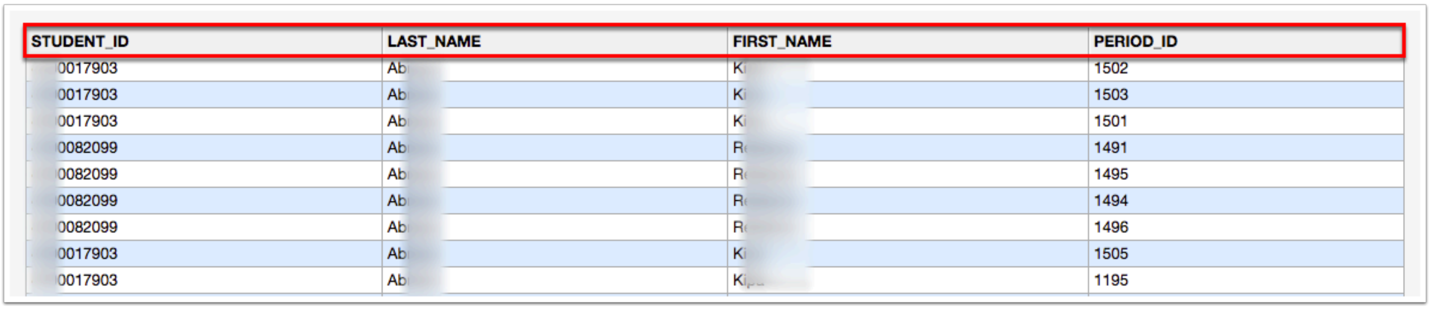
If the query results are paginated, click the page number to jump to the page or click the double arrows to jump to the last set of pages.



SELECT s.student_id, s.last_name, s.first_name, ap.period_id FROM students s JOIN student_enrollment se ON (s.stud
(s.student_id = ap.student_id) WHERE se.syear = 2022 AND se.end_date IS NULL

Time: 0.00928s **Records:** 2869609 **Showing:** 1 to 50

To sort results, you can click the headers. Click once for ascending results; click twice for descending results. In the image displayed, LAST_NAME has been clicked once.



STUDENT_ID	LAST_NAME	FIRST_NAME	PERIOD_ID
[redacted]0017903	Ab[redacted]	Ki[redacted]	1502
[redacted]0017903	Ab[redacted]	Ki[redacted]	1503
[redacted]0017903	Ab[redacted]	Ki[redacted]	1501
[redacted]0082099	Ab[redacted]	R[redacted]	1491
[redacted]0082099	Ab[redacted]	R[redacted]	1495
[redacted]0082099	Ab[redacted]	R[redacted]	1494
[redacted]0082099	Ab[redacted]	R[redacted]	1496
[redacted]0017903	Ab[redacted]	Ki[redacted]	1505
[redacted]0017903	Ab[redacted]	Ki[redacted]	1195