SQL 101

We will record today's presentation and provide the recording and slides after.

Introduce yourself in the Chat!

Name
Pronouns
Organization
What brought you here today?

Training Norms

1. Please keep yourself muted, help us reduce background noise.
2. Raise your hand or type questions into the chat.
3. Feel free to ask other participants questions in the chat!
4. We all have varying levels of Platform knowledge on this call. Ask the questions you have—no question is too simple or too advanced!

What is SQL?

What is SQL?

Structured Query Language (SQL)

- Used to query data in relational databases
- Search, update, and delete data from your database
- Several different types of SQL syntax that exist
  - Platform uses Amazon Redshift SQL
  - See AWS documentation for full list of possible commands https://docs.aws.amazon.com/redshift/latest/dg/cm_chap_SQLCommandRef.html

Example Dataset

Example Dataset

Sample table *public.sql_101* contains the following columns:

- gift_id
- constituent_id
- constituent_state
- gift_type
- gift_amount

Example Dataset

Sample table *public.sql_101*:

gift_id	constituent_id	constituent_state	gift_amount	gift_type
g-001R4i267	c-mnWWBX709	Florida	10	Monthly
g-001yz8698	c-E2hxJ1876	Nebraska	15	Monthly
g-0056Bk785	c-YP6jz5778	Pennsylvania	110	One-Off
g-006bdY402	c-PVSMW4806	Texas	20	Monthly
g-00AA6b694	c-18pXkP963	California	40	Monthly

SELECT

Select data from a table

SELECT

Selecting all columns in a table

```
SELECT *
FROM public.sql_101;
```

* SQL commands can be uppercase or lowercase - there are plenty of style guides available!

SELECT

Selecting specific columns in a table

```
SELECT constituent_id, constituent_state
FROM public.sql_101;
```

* SQL queries can be contained on one or more lines – they can be split wherever a blank is allowed

```
SELECT constituent_id
--, constituent_state
FROM public.sql_101;
```

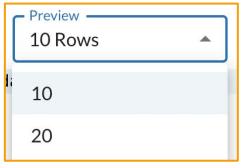
* Adding two dashes (--) before text will comment out the rest of the line

SELECT

Selecting a limited number of rows returned

```
SELECT *
FROM public.sql_101
LIMIT 30;
```

* Limit is different than preview rows. Preview rows dropdown affects your view of results, not the query that runs



SELECT

Selecting only distinct rows in a table

```
SELECT DISTINCT *
FROM public.sql_101;
```

SELECT

Selecting only distinct values of a column

```
SELECT DISTINCT constituent_state
FROM public.sql_101;
```

WHERE CONDITIONS

Apply a filter to the results of the query on one or more columns

WHERE

Add a filter on one column

```
SELECT
    *
FROM public.sql_101
WHERE constituent_state = 'Tennessee';
```

* SQL queries are sensitive to double vs. single quotes.

Single quotes are used to query specific values within the data.

Double quotes are used to differentiate things like column names from SQL functions, such as a column named “DATE” vs. the SQL function DATE.

WHERE

Add a filter on multiple columns

```
SELECT
    *
FROM public.sql_101
WHERE gift_type = 'Monthly'
AND gift_amount > 100;
```

WHERE

Include rows where either condition is true

```
SELECT
    *
FROM public.sql_101
WHERE gift_type = 'Monthly'
OR gift_amount > 100;
```

WHERE

Other WHERE condition operators available

`=`	Equal For booleans, 'is' or 'is not' may be used, ex. `employed is true`
`!= or <>`	Not equal
`<, >,` `<=, >=`	Less than, greater than, Less than or equal to, greater than or equal to
`between`	In an inclusive range, typically used for numeric or date ranges ex. `signup_date between '2014-01-01' and '2014-01-31'`
`in`	In a comma-separated list of values ex. `constituent_state in ('Illinois', 'Kentucky')`
`like`	Matching a pattern (case-sensitive); use one or more '%' to denote wildcards ex. `first_name like '%BOB%'` would match 'BOBBY', but not 'Bob'
`ilike`	Matching a pattern (case-insensitive); use one or more '%' to denote wildcards ex. `first_name ilike '%Bob%'` would match 'Bob', 'Bobby', 'Jim Bob', etc.

AGGREGATES

Performs a calculation, such as the average, on a numeric column

AGGREGATES

Count number of records

* If we want to give the data returned by our query a specific new column name, we can do that by defining the column name using `AS`. This is also referred to as an “alias.”

```
SELECT
    COUNT(*) AS count_rows,
    COUNT(DISTINCT constituent_state) AS count_states
FROM public.sql_101;
```

AGGREGATES

Sum of all values within a column

```
SELECT
    SUM(gift_amount) AS sum_gift
FROM public.sql_101;
```

AGGREGATES

Average of all values within a column

```
SELECT
    AVG(gift_amount) AS average_gift
FROM public.sql_101
WHERE gift_type = 'Monthly';
```

* If we combine an aggregate with a conditional clause, SQL will first filter the data, then apply the aggregation function to the filtered data set.

AGGREGATES

Minimum/Maximum value within a column

```
SELECT
    MIN(gift_amount) AS minimum_gift
FROM public.sql_101;
```

* You can also select the maximum value by using MAX(gift_amount)

GROUP BY

Group the results of an aggregate statement by another column

GROUP BY

Aggregate grouped by another column

```
SELECT
   constituent_state,
   AVG(gift_amount) AS average_gift
FROM public.sql_101
GROUP BY constituent_state;
```

ORDER BY

Sort the results of the query in ascending or descending order

ORDER BY

Sort results in descending order

* You can substitute DESC for ASC to receive the results in ascending order

```
SELECT
   constituent_state,
   AVG(gift_amount) AS average_gift
FROM public.sql_101
GROUP BY constituent_state
ORDER BY average_gift DESC
```

CASE WHEN

Returns a value based on conditions

CASE WHEN

Create a new field based on list of conditions

```
SELECT
    gift_id,
    CASE
        WHEN gift_amount < 50 THEN 'small'
        ELSE 'large'
    END AS gift_category
FROM public.sql_101
```

* If there is no ELSE and all conditions are false, returns NULL

INSERT

Insert new rows into a table

INSERT

Insert new rows into a table

```
INSERT INTO public.sql_101 VALUES
('g-001R4i436', 'c-mnWWBX709', 'Florida', 50, 'Monthly')
```

DELETE

Delete rows from a table

DELETE

Delete all rows which meet a certain condition

```
DELETE FROM public.sql_101
WHERE gift_id = 'g-001R4i436'
```

UPDATE

Change the value of certain rows

UPDATE

Update all rows which meet a certain condition

```
UPDATE public.sql_101
SET constituent_state = 'Indiana'
WHERE constituent_id = 'c-mnWWBX709'
```

PLATFORM QUERY DEMO

Thank you!

Reach out to **Support@CivisAnalytics.com** with any questions!