☑ **14. Formatting Best Practices**

☑ 15. Video: LIMIT

☑ 16. Quiz: LIMIT

☑ 17. Solution: LIMIT

☑ 18. Video: ORDER BY

☑ 19. Quiz: ORDER BY

○ 20. Solutions: ORDER BY

○ 21. Video: ORDER BY Part II

```
SELECT account_id
FROM orders
```

Phew!!! That was a lot of rules. Let's just write some queries. You will make mistakes, but that is part of the learning process!

Input   HISTORY ∨   MENU ∨

SCHEMA   ↺   1   SELECT id,name FROM sales_reps
                                    2
accounts   ∨

orders   ∨

region   ∨

sales_reps   ∨

web_events   ∨   EVALUATE

Output

SEARCH

RESOURCES ▲

CONCEPTS ▼

☑ **16. Quiz: LIMIT**

☑ 17. Solution: LIMIT

☑ 18. Video: ORDER BY

☑ 19. Quiz: ORDER BY

○ 20. Solutions: ORDER BY

○ 21. Video: ORDER BY Part II

○ 22. Quiz: ORDER BY Part II

○ 23. Solutions: ORDER BY Part II

Can You Use LIMIT?

1. Try using LIMIT yourself below by writing a query that displays all the data in the `occurred_at`, `account_id`, and `channel` columns of the web_events table, and limits the output to only the first 15 rows.

Input

HISTORY ∨   MENU ∨

SCHEMA ↻

accounts ∨

orders ∨

region ∨

sales_reps ∨

web_events ∨

```
1  SELECT id,account_id,occurred_at,channel FROM
   web_events
2  LIMIT 15;
```

EVALUATE

Output

SEARCH

RESOURCES ▲

CONCEPTS ▼

☑ **19. Quiz: ORDER BY**

● 20. Solutions: ORDER BY

● 21. Video: ORDER BY Part II

● 22. Quiz: ORDER BY Part II

● 23. Solutions: ORDER BY Part II

● 24. Video: WHERE

● 25. Quiz: WHERE

● 26. Solutions: WHERE

● 27. Video: WHERE with Non-Num...

● 28. Quiz: WHERE with Non-Nume...

● 29. Solutions: WHERE with Non-N...

● 30. Video: Arithmetic Operators

Practice

Let's get some practice using **ORDER BY**:

1. Write a query to return the 10 earliest orders in the **orders** table. Include the `id`, `occurred_at`, and `total_amt_usd`.

2. Write a query to return the top 5 **orders** in terms of largest `total_amt_usd`. Include the `id`, `account_id`, and `total_amt_usd`.

3. Write a query to return the lowest 20 **orders** in terms of smallest `total_amt_usd`. Include the `id`, `account_id`, and `total_amt_usd`.
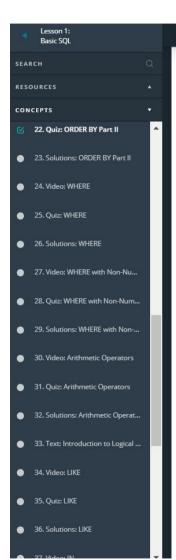
Remember, **just because a query successfully runs, does not mean you have the correct results.** To see if your query worked like you wanted it to, you need to examine your output to see if it satisfies the problem or the question. You can also compare your query and results with the solution on the following page (concept).
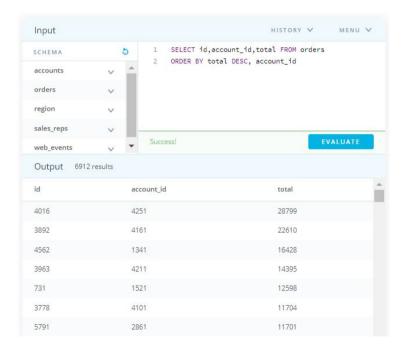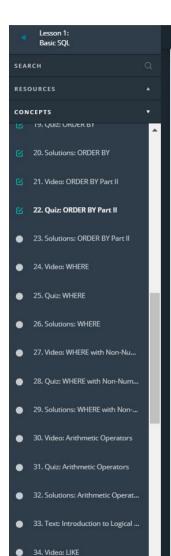
Input	HISTORY ∨	MENU ∨

SCHEMA ⟳	1	SELECT id,occurred_at,total_amt_usd FROM orders

poster_amt_usd

total_amt_usd

region ∨

sales_reps ∨

web_events ∨

Success!   **EVALUATE**

Output   6912 results

Practice

Let's get some practice using **ORDER BY**:

1. Write a query to return the 10 earliest orders in the **orders** table. Include the `id`, `occurred_at`, and `total_amt_usd`.

2. Write a query to return the top 5 **orders** in terms of largest `total_amt_usd`. Include the `id`, `account_id`, and `total_amt_usd`.

3. Write a query to return the lowest 20 **orders** in terms of smallest `total_amt_usd`. Include the `id`, `account_id`, and `total_amt_usd`.

Remember, **just because a query successfully runs, does not mean you have the correct results.** To see if your query worked like you wanted it to, you need to examine your output to see if it satisfies the problem or the question. You can also compare your query and results with the solution on the following page (concept).

Input		HISTORY ∨	MENU ∨

```
1   SELECT id,occurred_at,total_amt_usd FROM orders
2   ORDER By total_amt_usd DESC
3   LIMIT 5
```

SCHEMA

- accounts ∨
- orders ∨
- region ∨
- sales_reps ∨
- web_events ∨

Success!   **EVALUATE**

Output   5 results

id	occurred_at	total_amt_usd
4016	2016-12-26T08:53:24.000Z	232207.07
3892	2016-06-24T13:32:55.000Z	112875.18
3963	2015-03-30T00:05:30.000Z	107533.55
5791	2014-10-24T12:06:22.000Z	95005.82
3778	2016-07-17T14:50:43.000Z	93547.84

↑ Menu   ↗ Expand

Practice

Let's get some practice using **ORDER BY**:

1. Write a query to return the 10 earliest orders in the **orders** table. Include the `id`, `occurred_at`, and `total_amt_usd`.

2. Write a query to return the top 5 **orders** in terms of largest `total_amt_usd`. Include the `id`, `account_id`, and `total_amt_usd`.

3. Write a query to return the lowest 20 **orders** in terms of smallest `total_amt_usd`. Include the `id`, `account_id`, and `total_amt_usd`.

Remember, **just because a query successfully runs, does not mean you have the correct results.** To see if your query worked like you wanted it to, you need to examine your output to see if it satisfies the problem or the question. You can also compare your query and results with the solution on the following page (concept).

Input	HISTORY ∨ MENU ∨
SCHEMA	1 SELECT id,occurred_at,total_amt_usd FROM orders
accounts ∨	2 ORDER By total_amt_usd ASC
orders ∨	3 LIMIT 20
region ∨	
sales_reps ∨	
web_events ∨	Success! EVALUATE

Output   20 results

id	occurred_at	total_amt_usd
6375	2016-10-15T16:08:41.000Z	0.00
6435	2016-07-10T20:56:38.000Z	0.00
6312	2016-04-07T14:44:01.000Z	0.00
6323	2016-02-19T19:33:03.000Z	0.00
4446	2016-03-27T11:35:30.000Z	0.00
6009	2014-03-05T13:30:35.000Z	0.00
4625	2014-11-21T10:21:58.000Z	0.00
5612	2015-07-02T09:26:49.000Z	0.00

↑ Menu   ↗ Expand

SEARCH

RESOURCES ▲

CONCEPTS ▼

☑ 22. Quiz: ORDER BY Part II

● 23. Solutions: ORDER BY Part II

● 24. Video: WHERE

● 25. Quiz: WHERE

● 26. Solutions: WHERE

● 27. Video: WHERE with Non-Nu...

● 28. Quiz: WHERE with Non-Num...

● 29. Solutions: WHERE with Non-...

● 30. Video: Arithmetic Operators

● 31. Quiz: Arithmetic Operators

● 32. Solutions: Arithmetic Operat...

● 33. Text: Introduction to Logical ...

● 34. Video: LIKE

● 35. Quiz: LIKE

● 36. Solutions: LIKE

● 37. Video: IN

Questions

1. Write a query that displays the order ID, account ID, and total dollar amount for all the orders, sorted first by the account ID (in ascending order), and then by the total dollar amount (in descending order).

2. Now write a query that again displays order ID, account ID, and total dollar amount for each order, but this time sorted first by total dollar amount (in descending order), and then by account ID (in ascending order).

3. Compare the results of these two queries above. How are the results different when you switch the column you sort on first?
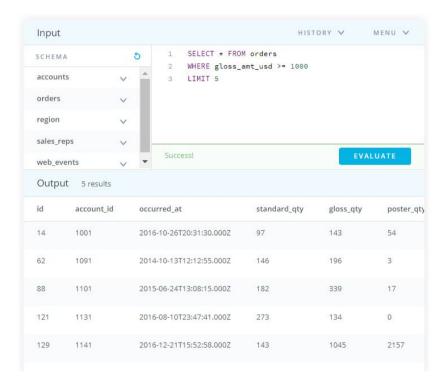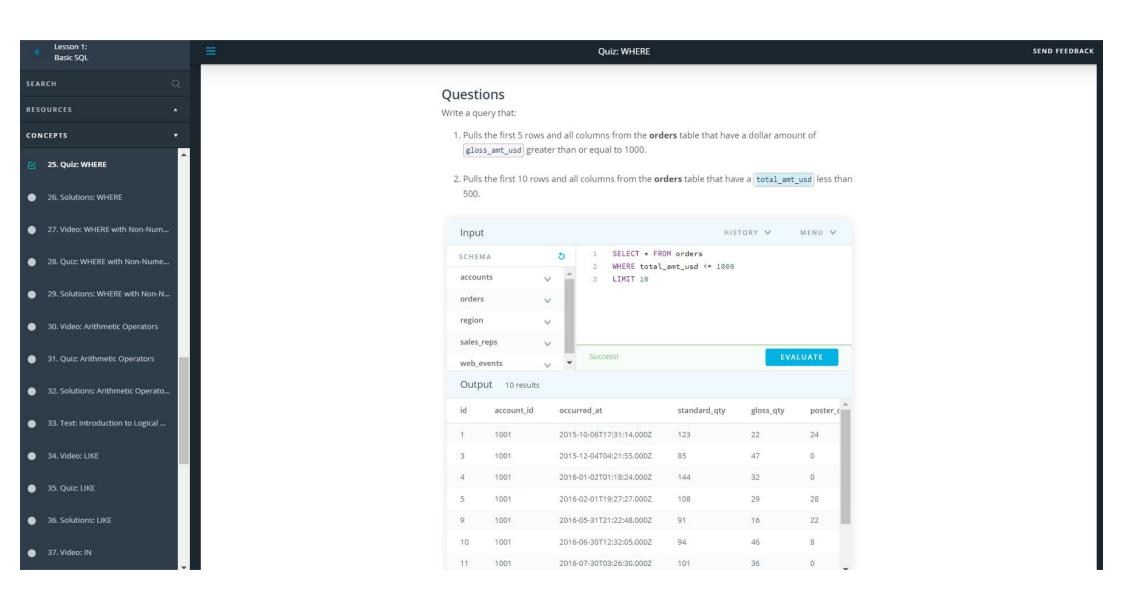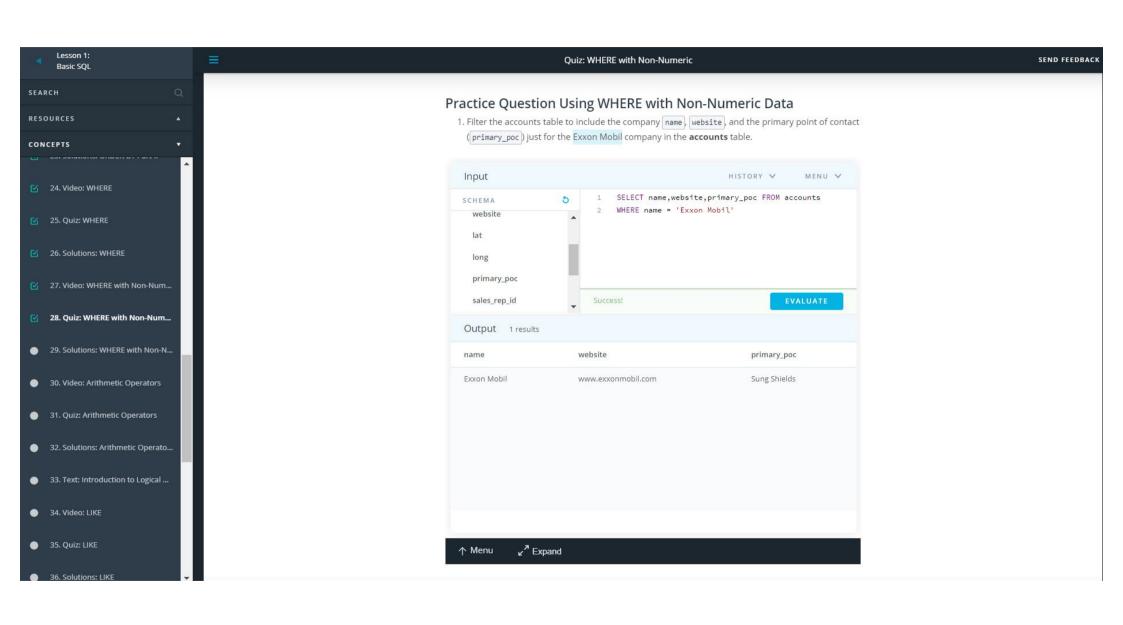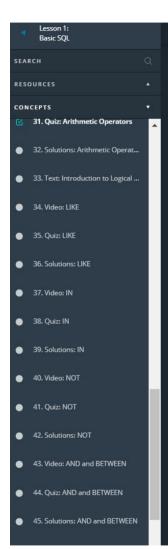
Input		HISTORY ∨	MENU ∨

SCHEMA	↻
accounts	∨
orders	∨
region	∨
sales_reps	∨
web_events	∨

```
1   SELECT id,account_id,total FROM orders
2   ORDER BY total DESC, account_id
```

Success!   EVALUATE

Output   6912 results

id	account_id	total
4016	4251	28799
3892	4161	22610
4562	1341	16428
3963	4211	14395
731	1521	12598
3778	4101	11704
5791	2861	11701

Questions

1. Write a query that displays the order ID, account ID, and total dollar amount for all the orders, sorted first by the account ID (in ascending order), and then by the total dollar amount (in descending order).

2. Now write a query that again displays order ID, account ID, and total dollar amount for each order, but this time sorted first by total dollar amount (in descending order), and then by account ID (in ascending order).

3. Compare the results of these two queries above. How are the results different when you switch the column you sort on first?

Input		HISTORY ∨	MENU ∨

```
SCHEMA          ↻        1   SELECT id,account_id,total FROM orders
                         2   ORDER BY account_id,total DESC
accounts        ∨

orders          ∨

region          ∨

sales_reps      ∨

web_events      ∨
```

Success!   EVALUATE

Output   6912 results

id	account_id	total
4308	1001	1410
4309	1001	1405
4316	1001	1384
4317	1001	1347
4314	1001	1343
4307	1001	1321
4311	1001	1307

SEARCH

RESOURCES ▲

CONCEPTS ▼

Questions

Write a query that:

1. Pulls the first 5 rows and all columns from the **orders** table that have a dollar amount of `gloss_amt_usd` greater than or equal to 1000.

2. Pulls the first 10 rows and all columns from the **orders** table that have a `total_amt_usd` less than 500.

Input			HISTORY ∨	MENU ∨

SCHEMA	↻
accounts	∨
orders	∨
region	∨
sales_reps	∨
web_events	∨

```
1   SELECT * FROM orders
2   WHERE gloss_amt_usd >= 1000
3   LIMIT 5
```

Success!   EVALUATE

Output   5 results

id	account_id	occurred_at	standard_qty	gloss_qty	poster_qty
14	1001	2016-10-26T20:31:30.000Z	97	143	54
62	1091	2014-10-13T12:12:55.000Z	146	196	3
88	1101	2015-06-24T13:08:15.000Z	182	339	17
121	1131	2016-08-10T23:47:41.000Z	273	134	0
129	1141	2016-12-21T15:52:58.000Z	143	1045	2157

Questions

Write a query that:

1. Pulls the first 5 rows and all columns from the **orders** table that have a dollar amount of `gloss_amt_usd` greater than or equal to 1000.

2. Pulls the first 10 rows and all columns from the **orders** table that have a `total_amt_usd` less than 500.

Input		HISTORY ∨	MENU ∨

SCHEMA	↻
accounts	∨
orders	∨
region	∨
sales_reps	∨
web_events	∨

```
1   SELECT * FROM orders
2   WHERE total_amt_usd <= 1000
3   LIMIT 10
```

Success!   **EVALUATE**

Output   10 results

id	account_id	occurred_at	standard_qty	gloss_qty	poster_q
1	1001	2015-10-06T17:31:14.000Z	123	22	24
3	1001	2015-12-04T04:21:55.000Z	85	47	0
4	1001	2016-01-02T01:18:24.000Z	144	32	0
5	1001	2016-02-01T19:27:27.000Z	108	29	28
9	1001	2016-05-31T21:22:48.000Z	91	16	22
10	1001	2016-06-30T12:32:05.000Z	94	46	8
11	1001	2016-07-30T03:26:30.000Z	101	36	0

Practice Question Using WHERE with Non-Numeric Data

1. Filter the accounts table to include the company `name`, `website`, and the primary point of contact (`primary_poc`) just for the Exxon Mobil company in the **accounts** table.

Input

HISTORY ∨   MENU ∨

SCHEMA

website
lat
long
primary_poc
sales_rep_id

```
1   SELECT name,website,primary_poc FROM accounts
2   WHERE name = 'Exxon Mobil'
```

Success!   **EVALUATE**

Output   1 results

name	website	primary_poc
Exxon Mobil	www.exxonmobil.com	Sung Shields

↑ Menu   ↗ Expand

Questions using Arithmetic Operations

Using the **orders** table:

1. Create a column that divides the `standard_amt_usd` by the `standard_qty` to find the unit price for standard paper for each order. Limit the results to the first 10 orders, and include the `id` and `account_id` fields.

2. Write a query that finds the percentage of revenue that comes from poster paper for each order. You will need to use only the columns that end with `_usd`. (Try to do this without using the `total` column.) Display the `id` and `account_id` fields also. **NOTE - you will receive an error with the correct solution to this question. This occurs because at least one of the values in the data creates a division by zero in your formula. You will learn later in the course how to fully handle this issue. For now, you can just limit your calculations to the first 10 orders, as we did in question #1, and you'll avoid that set of data that causes the problem.**

Notice, the above operators combine information across columns for the same row. If you want to combine values of a particular column, across multiple rows, we will do this with aggregations. We will get to that before the end of the course!

Input	HISTORY ⌄	MENU ⌄

SCHEMA ↻		
accounts ⌄		
orders ⌄		
region ⌄		
sales_reps ⌄		
web_events ⌄		

```
1   SELECT id,account_id,
        (standard_amt_usd/standard_qty) AS unit_price
2   FROM orders
3   LIMIT 10
```

Success!   **EVALUATE**

Output   10 results

id	account_id	unit_price
1	1001	4.9900000000000000
2	1001	4.9900000000000000
3	1001	4.9000000000000000

SEARCH

RESOURCES ▲

CONCEPTS ▼

Questions using the LIKE operator

Use the **accounts** table to find

    1. All the companies whose names start with 'C'.

    2. All companies whose names contain the string 'one' somewhere in the name.

    3. All companies whose names end with 's'.
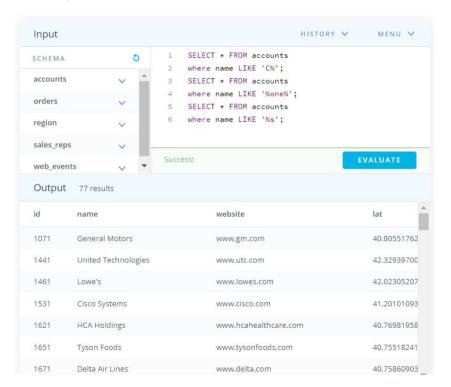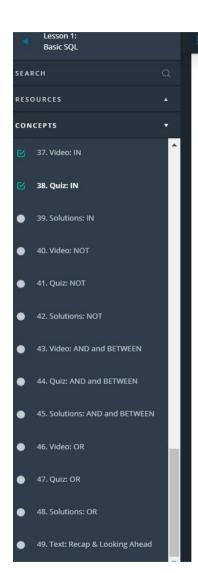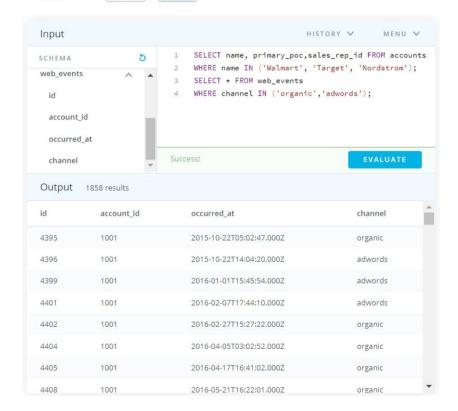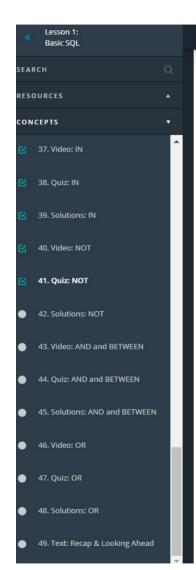
Input	HISTORY ⌄	MENU ⌄

SCHEMA ↻	

```
1   SELECT * FROM accounts
2   where name LIKE 'C%'
```

accounts ⌃

  id

  name

  website

  lat

Success!   EVALUATE

Output   37 results

id	name	website	lat
1061	CVS Health	www.cvshealth.com	41.46779585
1131	Chevron	www.chevron.com	42.61194130
1141	Costco	www.costco.com	42.26304566
1201	Cardinal Health	www.cardinal.com	41.31906537
1281	Citigroup	www.citigroup.com	40.85305429
1361	Comcast	www.comcastcorporation.com	42.54154764
1531	Cisco Systems	www.cisco.com	41.20101093

Questions using the LIKE operator

Use the **accounts** table to find

1. All the companies whose names start with 'C'.

2. All companies whose names contain the string 'one' somewhere in the name.

3. All companies whose names end with 's'.

Input		HISTORY ⌄	MENU ⌄

SCHEMA ↻	
accounts ⌄	
orders ⌄	
region ⌄	
sales_reps ⌄	
web_events ⌄	

```
1  SELECT * FROM accounts
2  where name LIKE 'C%';
3  SELECT * FROM accounts
4  where name LIKE '%one%';
5  SELECT * FROM accounts
6  where name LIKE '%s';
```

Success!   **EVALUATE**

Output   77 results

id	name	website	lat
1071	General Motors	www.gm.com	40.80551762
1441	United Technologies	www.utc.com	42.32939700
1461	Lowe's	www.lowes.com	42.02305207
1531	Cisco Systems	www.cisco.com	41.20101093
1621	HCA Holdings	www.hcahealthcare.com	40.76981958
1651	Tyson Foods	www.tysonfoods.com	40.75518241
1671	Delta Air Lines	www.delta.com	40.75860903

SEARCH

RESOURCES ▲

CONCEPTS ▼

☑ 37. Video: IN

☑ **38. Quiz: IN**

● 39. Solutions: IN

● 40. Video: NOT

● 41. Quiz: NOT

● 42. Solutions: NOT

● 43. Video: AND and BETWEEN

● 44. Quiz: AND and BETWEEN

● 45. Solutions: AND and BETWEEN

● 46. Video: OR

● 47. Quiz: OR

● 48. Solutions: OR

● 49. Text: Recap & Looking Ahead

Questions using IN operator

1. Use the **accounts** table to find the account `name`, `primary_poc`, and `sales_rep_id` for Walmart, Target, and Nordstrom.

2. Use the **web_events** table to find all information regarding individuals who were contacted via the **channel** of `organic` or `adwords`.

Input	HISTORY ∨	MENU ∨

SCHEMA ↻	
web_events ∧	
id	
account_id	
occurred_at	
channel	

```
1  SELECT name, primary_poc,sales_rep_id FROM accounts
2  WHERE name IN ('Walmart', 'Target', 'Nordstrom');
3  SELECT * FROM web_events
4  WHERE channel IN ('organic','adwords');
```

Success!   **EVALUATE**

Output   1858 results

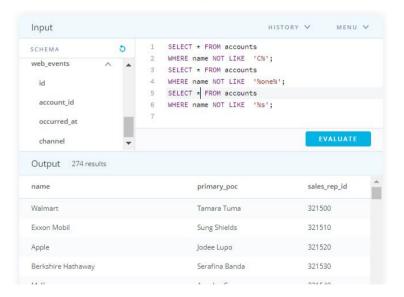| id	account_id	occurred_at	channel
4395	1001	2015-10-22T05:02:47.000Z	organic
4396	1001	2015-10-22T14:04:20.000Z	adwords
4399	1001	2016-01-01T15:45:54.000Z	adwords
4401	1001	2016-02-07T17:44:10.000Z	adwords
4402	1001	2016-02-27T15:27:22.000Z	organic
4404	1001	2016-04-05T03:02:52.000Z	organic
4405	1001	2016-04-17T16:41:02.000Z	organic
4408	1001	2016-05-21T16:22:01.000Z	organic

Questions using the NOT operator

We can pull all of the rows that were excluded from the queries in the previous two concepts with our new operator.

1. Use the **accounts** table to find the account name, primary poc, and sales rep id for all stores except Walmart, Target, and Nordstrom.

2. Use the **web_events** table to find all information regarding individuals who were contacted via any method except using `organic` or `adwords` methods.
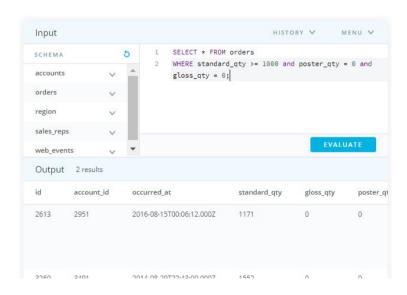
Use the **accounts** table to find:

1. All the companies whose names do not start with 'C'.

2. All companies whose names do not contain the string 'one' somewhere in the name.

3. All companies whose names do not end with 's'.

Input

HISTORY ∨   MENU ∨

SCHEMA ↻

web_events ∧

  id

  account_id

  occurred_at

  channel

```
1    SELECT name,primary_poc,sales_rep_id FROM accounts
2    WHERE name NOT IN ('Walmart', 'Target','Nordstrom');
3    SELECT * FROM web_events
4    WHERE channel NOT IN ('organic','adwords');
```

Success!   **EVALUATE**

Output   7215 results

id	account_id	occurred_at	channel
1	1001	2015-10-06T17:13:58.000Z	direct

Questions using the NOT operator

We can pull all of the rows that were excluded from the queries in the previous two concepts with our new operator.

1. Use the **accounts** table to find the account name, primary poc, and sales rep id for all stores except Walmart, Target, and Nordstrom.

2. Use the **web_events** table to find all information regarding individuals who were contacted via any method except using `organic` or `adwords` methods.

Use the **accounts** table to find:

1. All the companies whose names do not start with 'C'.

2. All companies whose names do not contain the string 'one' somewhere in the name.

3. All companies whose names do not end with 's'.

Input		HISTORY ∨	MENU ∨

SCHEMA	↻		

web_events ∧

  id

  account_id

  occurred_at

  channel

```
1  SELECT * FROM accounts
2  WHERE name NOT LIKE  'C%';
3  SELECT * FROM accounts
4  WHERE name NOT LIKE  '%one%';
5  SELECT * FROM accounts
6  WHERE name NOT LIKE  '%s';
7
```

EVALUATE

Output   274 results

name	primary_poc	sales_rep_id
Walmart	Tamara Tuma	321500
Exxon Mobil	Sung Shields	321510
Apple	Jodee Lupo	321520
Berkshire Hathaway	Serafina Banda	321530

SEARCH

RESOURCES ▲

CONCEPTS ▼

Questions using AND and BETWEEN operators

1. Write a query that returns all the **orders** where the `standard_qty` is over 1000, the `poster_qty` is 0, and the `gloss_qty` is 0.
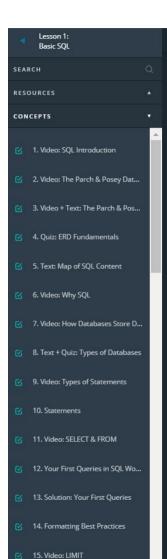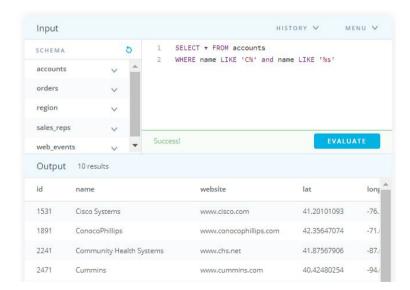
2. Using the **accounts** table, find all the companies whose names do not start with 'C' and end with 's'.

3. When you use the BETWEEN operator in SQL, do the results include the values of your endpoints, or not? Figure out the answer to this important question by writing a query that displays the order date and `gloss_qty` data for all **orders** where gloss_qty is between 24 and 29. Then look at your output to see if the BETWEEN operator included the begin and end values or not.

4. Use the **web_events** table to find all information regarding individuals who were contacted via the `organic` or `adwords` channels, and started their account at any point in 2016, sorted from newest to oldest.

Input	HISTORY ∨	MENU ∨

SCHEMA	↻
accounts	∨
orders	∨
region	∨
sales_reps	∨
web_events	∨

```
1   SELECT * FROM orders
2   WHERE standard_qty >= 1000 and poster_qty = 0 and
    gloss_qty = 0;
```

EVALUATE

Output   2 results

id	account_id	occurred_at	standard_qty	gloss_qty	poster_qt
2613	2951	2016-08-15T00:06:12.000Z	1171	0	0
3260	3401	2014-08-20T22:43:00.000Z	1552	0	0

SEARCH

RESOURCES ▲

CONCEPTS ▼

☑ 1. Video: SQL Introduction

☑ 2. Video: The Parch & Posey Dat...

☑ 3. Video + Text: The Parch & Pos...

☑ 4. Quiz: ERD Fundamentals

☑ 5. Text: Map of SQL Content

☑ 6. Video: Why SQL

☑ 7. Video: How Databases Store D...

☑ 8. Text + Quiz: Types of Databases

☑ 9. Video: Types of Statements

☑ 10. Statements

☑ 11. Video: SELECT & FROM

☑ 12. Your First Queries in SQL Wo...

☑ 13. Solution: Your First Queries

☑ 14. Formatting Best Practices

☑ 15. Video: LIMIT

Questions using AND and BETWEEN operators

1. Write a query that returns all the **orders** where the `standard_qty` is over 1000, the `poster_qty` is 0, and the `gloss_qty` is 0.

2. Using the **accounts** table, find all the companies whose names do not start with 'C' and end with 's'.

3. When you use the BETWEEN operator in SQL, do the results include the values of your endpoints, or not? Figure out the answer to this important question by writing a query that displays the order date and `gloss_qty` data for all **orders** where gloss_qty is between 24 and 29. Then look at your output to see if the BETWEEN operator included the begin and end values or not.

4. Use the **web_events** table to find all information regarding individuals who were contacted via the `organic` or `adwords` channels, and started their account at any point in 2016, sorted from newest to oldest.

Input	HISTORY ∨	MENU ∨

SCHEMA ↻	
accounts	∨
orders	∨
region	∨
sales_reps	∨
web_events	∨

```
1    SELECT * FROM accounts
2    WHERE name LIKE 'C%' and name LIKE '%s'
```

Success!   EVALUATE

Output   10 results

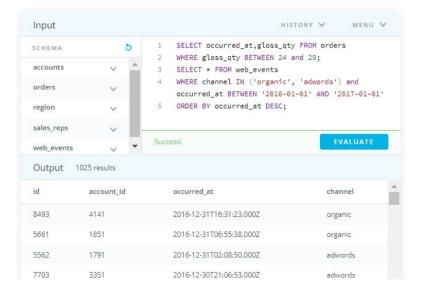| id	name	website	lat	long
1531	Cisco Systems	www.cisco.com	41.20101093	-76.
1891	ConocoPhillips	www.conocophillips.com	42.35647074	-71.
2241	Community Health Systems	www.chs.net	41.87567906	-87.
2471	Cummins	www.cummins.com	40.42480254	-94.

Questions using AND and BETWEEN operators

1. Write a query that returns all the **orders** where the `standard_qty` is over 1000, the `poster_qty` is 0, and the `gloss_qty` is 0.

2. Using the **accounts** table, find all the companies whose names do not start with 'C' and end with 's'.

3. When you use the BETWEEN operator in SQL, do the results include the values of your endpoints, or not? Figure out the answer to this important question by writing a query that displays the order date and `gloss_qty` data for all **orders** where gloss_qty is between 24 and 29. Then look at your output to see if the BETWEEN operator included the begin and end values or not.

4. Use the **web_events** table to find all information regarding individuals who were contacted via the `organic` or `adwords` channels, and started their account at any point in 2016, sorted from newest to oldest.

Input   HISTORY ∨   MENU ∨

SCHEMA	↻
accounts	∨
orders	∨
region	∨
sales_reps	∨
web_events	∨

```
1  SELECT occurred_at,gloss_qty FROM orders
2  WHERE gloss_qty BETWEEN 24 and 29;
3  SELECT * FROM web_events
4  WHERE channel IN ('organic', 'adwords') and
      occurred_at BETWEEN '2016-01-01' AND '2017-01-01'
5  ORDER BY occurred_at DESC;
```

Success!   **EVALUATE**

Output   1025 results

id	account_id	occurred_at	channel
8493	4141	2016-12-31T16:31:23.000Z	organic
5661	1851	2016-12-31T06:55:38.000Z	organic
5562	1791	2016-12-31T02:08:50.000Z	adwords
7703	3351	2016-12-30T21:06:53.000Z	adwords

SEARCH

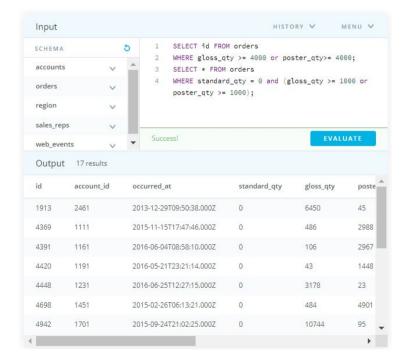RESOURCES ▲

CONCEPTS ▼

Questions using the OR operator

1. Find list of **orders** ids where either `gloss_qty` or `poster_qty` is greater than 4000. Only include the `id` field in the resulting table.

2. Write a query that returns a list of **orders** where the `standard_qty` is zero and either the `gloss_qty` or `poster_qty` is over 1000.

3. Find all the company names that start with a 'C' or 'W', and the primary contact **contains** 'ana' or 'Ana', but it doesn't contain 'eana'.

Input		HISTORY ∨	MENU ∨

SCHEMA	↻

```
1   SELECT id FROM orders
2   WHERE gloss_qty >= 4000 or poster_qty>= 4000;
3   SELECT * FROM orders
4   WHERE standard_qty = 0 and (gloss_qty >= 1000 or
        poster_qty >= 1000);
```

accounts ∨

orders ∨

region ∨

sales_reps ∨

web_events ∨

Success!   EVALUATE

Output   17 results

id	account_id	occurred_at	standard_qty	gloss_qty	poste
1913	2461	2013-12-29T09:50:38.000Z	0	6450	45
4369	1111	2015-11-15T17:47:46.000Z	0	486	2988
4391	1161	2016-06-04T08:58:10.000Z	0	106	2967
4420	1191	2016-05-21T23:21:14.000Z	0	43	1448
4448	1231	2016-06-25T12:27:15.000Z	0	3178	23
4698	1451	2015-02-26T06:13:21.000Z	0	484	4901
4942	1701	2015-09-24T21:02:25.000Z	0	10744	95

Questions using the OR operator

1. Find list of **orders** ids where either `gloss_qty` or `poster_qty` is greater than 4000. Only include the `id` field in the resulting table.

2. Write a query that returns a list of **orders** where the `standard_qty` is zero and either the `gloss_qty` or `poster_qty` is over 1000.

3. Find all the company names that start with a 'C' or 'W', and the primary contact **contains** 'ana' or 'Ana', but it doesn't contain 'eana'.

Input	HISTORY ∨	MENU ∨

SCHEMA	↻

```
1    SELECT * FROM accounts
2    WHERE (name LIKE 'C%' or name LIKE 'W%') and
         (primary_poc LIKE '%ana%'  or primary_poc LIKE
         '%Ana%') and primary_poc NOT LIKE '%eana%'
3
```

lat

long

primary_poc

sales_rep_id

orders

Success!   **EVALUATE**

Output   2 results

id	name	website	lat	long	pr
1061	CVS Health	www.cvshealth.com	41.46779585	-73.76763638	Ar
1361	Comcast	www.comcastcorporation.com	42.54154764	-76.24992387	Sh