Filtering Data

Let us understand how we can filter the data as part of our queries.

- We use `WHERE` clause to filter the data.
- All comparison operators such as `=, !=, >, <, <=, >=` etc can be used to compare a column or expression or literal with another column or expression or literal.
- We can use operators such as `LIKE` with `%` or `~` with regular expressions for pattern matching.
- Boolean `OR` and `AND` can be performed when we want to apply multiple conditions.
    - Get all orders with order_status equals to COMPLETE or CLOSED. We can also use IN operator.
    - Get all orders from month 2014 January with order_status equals to COMPLETE or CLOSED
- We can also use `BETWEEN` along with `AND` to compare a column or expression against range of values.
- We need to use `IS NULL` and `IS NOT NULL` to compare against null values.

```
%load_ext sql
```

```
The sql extension is already loaded. To reload it, use:
  %reload_ext sql
```

```
%env
DATABASE_URL=postgresql://itversity_retail_user:retail_password@localhost:5432/itversity_retail
_db
```

```
env:
DATABASE_URL=postgresql://itversity_retail_user:retail_password@localhost:5432/itversity_reta
il_db
```

```
%%sql

SELECT * FROM orders
WHERE order_status = 'COMPLETE'
LIMIT 10
```

```
 * postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
10 rows affected.
```

order_id	order_date	order_customer_id	order_status
3	2013-07-25 00:00:00	12111	COMPLETE
5	2013-07-25 00:00:00	11318	COMPLETE
6	2013-07-25 00:00:00	7130	COMPLETE
7	2013-07-25 00:00:00	4530	COMPLETE
15	2013-07-25 00:00:00	2568	COMPLETE
17	2013-07-25 00:00:00	2667	COMPLETE
22	2013-07-25 00:00:00	333	COMPLETE
26	2013-07-25 00:00:00	7562	COMPLETE
28	2013-07-25 00:00:00	656	COMPLETE
32	2013-07-25 00:00:00	3960	COMPLETE

```
%sql SELECT count(1) FROM orders
```

```
 * postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
1 rows affected.
```

count

68883

```sql
%%sql

SELECT count(1)
FROM orders
WHERE order_status = 'COMPLETE'
```

count
22899

```sql
%%sql

SELECT DISTINCT order_status
FROM orders
WHERE order_status = 'COMPLETE'
```

order_status
COMPLETE

```sql
%%sql

SELECT DISTINCT order_status
FROM orders
```

order_status
COMPLETE
ON_HOLD
PENDING_PAYMENT
PENDING
CLOSED
CANCELED
PROCESSING
PAYMENT_REVIEW
SUSPECTED_FRAUD

```sql
%%sql

SELECT * FROM orders
WHERE order_status IN ('COMPLETE', 'CLOSED')
LIMIT 10
```

order_id	order_date	order_customer_id	order_status
1	2013-07-25 00:00:00	11599	CLOSED
3	2013-07-25 00:00:00	12111	COMPLETE
4	2013-07-25 00:00:00	8827	CLOSED
5	2013-07-25 00:00:00	11318	COMPLETE
6	2013-07-25 00:00:00	7130	COMPLETE
7	2013-07-25 00:00:00	4530	COMPLETE
12	2013-07-25 00:00:00	1837	CLOSED
15	2013-07-25 00:00:00	2568	COMPLETE
17	2013-07-25 00:00:00	2667	COMPLETE
18	2013-07-25 00:00:00	1205	CLOSED

```
%%sql

SELECT count(1) FROM orders
WHERE order_status IN ('COMPLETE', 'CLOSED')
```

* postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
1 rows affected.

count

30455

```
%%sql

SELECT count(1) FROM orders
WHERE order_status = 'COMPLETE' OR order_status = 'CLOSED'
```

* postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
1 rows affected.

count

30455

```
%%sql

SELECT * FROM orders
WHERE order_date = '2014-01-01'
LIMIT 3
```

* postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
3 rows affected.

order_id	order_date	order_customer_id	order_status
25876	2014-01-01 00:00:00	3414	PENDING_PAYMENT
25877	2014-01-01 00:00:00	5549	PENDING_PAYMENT
25878	2014-01-01 00:00:00	9084	PENDING

ℹ **Note**

This query will not work as LIKE cannot be used to compare against columns with date data type

```
%%sql

SELECT * FROM orders
WHERE order_date LIKE '2014-01%'
LIMIT 3
```

```
  * postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
(psycopg2.errors.UndefinedFunction) operator does not exist: timestamp without time zone ~~
unknown
LINE 2: WHERE order_date LIKE '2014-01%'
                                        ^
HINT:  No operator matches the given name and argument types. You might need to add explicit
type casts.

[SQL: SELECT * FROM orders
WHERE order_date LIKE '2014-01%%'
LIMIT 3]
(Background on this error at: http://sqlalche.me/e/13/f405)
```

%%**sql**

```
SELECT * FROM orders
WHERE order_status IN ('COMPLETE', 'CLOSED')
    AND to_char(order_date, 'yyyy-MM-dd') LIKE '2014-01%'
LIMIT 10
```

```
  * postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
10 rows affected.
```

order_id	order_date	order_customer_id	order_status
25882	2014-01-01 00:00:00	4598	COMPLETE
25888	2014-01-01 00:00:00	6735	COMPLETE
25889	2014-01-01 00:00:00	10045	COMPLETE
25891	2014-01-01 00:00:00	3037	CLOSED
25895	2014-01-01 00:00:00	1044	COMPLETE
25897	2014-01-01 00:00:00	6405	COMPLETE
25898	2014-01-01 00:00:00	3950	COMPLETE
25899	2014-01-01 00:00:00	8068	CLOSED
25900	2014-01-01 00:00:00	2382	CLOSED
25901	2014-01-01 00:00:00	3099	COMPLETE

%%**sql**

```
SELECT count(1) FROM orders
WHERE order_status IN ('COMPLETE', 'CLOSED')
    AND to_char(order_date, 'yyyy-MM-dd') LIKE '2014-01%'
```

```
  * postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
1 rows affected.
```

count
2544

%%**sql**

```
SELECT * FROM orders
WHERE order_status IN ('COMPLETE', 'CLOSED')
    AND to_char(order_date, 'yyyy-MM') = '2014-01'
LIMIT 10
```

```
  * postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
10 rows affected.
```

order_id	order_date	order_customer_id	order_status
25882	2014-01-01 00:00:00	4598	COMPLETE
25888	2014-01-01 00:00:00	6735	COMPLETE
25889	2014-01-01 00:00:00	10045	COMPLETE
25891	2014-01-01 00:00:00	3037	CLOSED
25895	2014-01-01 00:00:00	1044	COMPLETE
25897	2014-01-01 00:00:00	6405	COMPLETE
25898	2014-01-01 00:00:00	3950	COMPLETE
25899	2014-01-01 00:00:00	8068	CLOSED
25900	2014-01-01 00:00:00	2382	CLOSED
25901	2014-01-01 00:00:00	3099	COMPLETE

```sql
%%sql

SELECT count(1) FROM orders
WHERE order_status IN ('COMPLETE', 'CLOSED')
    AND to_char(order_date, 'yyyy-MM') = '2014-01'
```

 * postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
1 rows affected.

count
2544

```sql
%%sql

SELECT count(1) FROM orders
WHERE order_status IN ('COMPLETE', 'CLOSED')
    AND to_char(order_date, 'yyyy-MM-dd') ~ '2014-01'
```

 * postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
1 rows affected.

count
2544

```sql
%%sql

SELECT count(1), min(order_date), max(order_date), count(DISTINCT order_date)
FROM orders
WHERE order_status IN ('COMPLETE', 'CLOSED')
    AND order_date BETWEEN '2014-01-01' AND '2014-03-31'
```

 * postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
1 rows affected.

count	min	max	count_1
7594	2014-01-01 00:00:00	2014-03-31 00:00:00	89

```sql
%%sql

SELECT DISTINCT order_date
FROM orders
WHERE to_char(order_date, 'yyyy-MM') LIKE '2014-03%'
ORDER BY order_date
```

 * postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
30 rows affected.

order_date
2014-03-01 00:00:00
2014-03-02 00:00:00
2014-03-03 00:00:00
2014-03-04 00:00:00
2014-03-05 00:00:00
2014-03-06 00:00:00
2014-03-07 00:00:00
2014-03-08 00:00:00
2014-03-10 00:00:00
2014-03-11 00:00:00
2014-03-12 00:00:00
2014-03-13 00:00:00
2014-03-14 00:00:00
2014-03-15 00:00:00
2014-03-16 00:00:00
2014-03-17 00:00:00
2014-03-18 00:00:00
2014-03-19 00:00:00
2014-03-20 00:00:00
2014-03-21 00:00:00
2014-03-22 00:00:00
2014-03-23 00:00:00
2014-03-24 00:00:00
2014-03-25 00:00:00
2014-03-26 00:00:00
2014-03-27 00:00:00
2014-03-28 00:00:00
2014-03-29 00:00:00
2014-03-30 00:00:00
2014-03-31 00:00:00

```
%%sql

DROP TABLE IF EXISTS users
```

 * postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
Done.

```
[]
```

```
%%sql

CREATE TABLE users (
    user_id SERIAL PRIMARY KEY,
    user_first_name VARCHAR(30) NOT NULL,
    user_last_name VARCHAR(30) NOT NULL,
    user_email_id VARCHAR(50) NOT NULL,
    user_email_validated BOOLEAN DEFAULT FALSE,
    user_password VARCHAR(200),
    user_role VARCHAR(1) NOT NULL DEFAULT 'U', --U and A
    is_active BOOLEAN DEFAULT FALSE,
    create_ts TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    last_updated_ts TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```
  * postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
Done.
```

```
[]
```

```
%%sql

INSERT INTO users (user_first_name, user_last_name, user_email_id)
VALUES ('Donald', 'Duck', 'donald@duck.com')
```

```
  * postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
1 rows affected.
```

```
[]
```

```
%%sql

INSERT INTO users (user_first_name, user_last_name, user_email_id, user_role, is_active)
VALUES ('Mickey', 'Mouse', 'mickey@mouse.com', 'U', true)
```

```
  * postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
1 rows affected.
```

```
[]
```

```
%%sql

INSERT INTO users
    (user_first_name, user_last_name, user_email_id, user_password, user_role, is_active)
VALUES
    ('Gordan', 'Bradock', 'gbradock0@barnesandnoble.com', 'h9LAz7p7ub', 'U', true),
    ('Tobe', 'Lyness', 'tlyness1@paginegialle.it', 'oEofndp', 'U', true),
    ('Addie', 'Mesias', 'amesias2@twitpic.com', 'ih7Y69u56', 'U', true)
```

```
  * postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
3 rows affected.
```

```
[]
```

```
%%sql

SELECT * FROM users
```

```
  * postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
5 rows affected.
```

user_id	user_first_name	user_last_name	user_email_id	user_email_validated
1	Donald	Duck	donald@duck.com	False
2	Mickey	Mouse	mickey@mouse.com	False
3	Gordan	Bradock	gbradock0@barnesandnoble.com	False
4	Tobe	Lyness	tlyness1@paginegialle.it	False
5	Addie	Mesias	amesias2@twitpic.com	False

ⓘ Note

This will not return any thing and not the correct way to compare against NULL. NULL is specially treated by databases and it is not same as empty string.

```
%%sql

SELECT * FROM users
WHERE user_password = NULL
```

 * postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
0 rows affected.

user_id	user_first_name	user_last_name	user_email_id	user_email_validated	user_password	

```
%%sql

SELECT * FROM users
WHERE user_password IS NULL
```

 * postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
2 rows affected.

user_id	user_first_name	user_last_name	user_email_id	user_email_validated	user_passw
1	Donald	Duck	donald@duck.com	False	N
2	Mickey	Mouse	mickey@mouse.com	False	N

```
%%sql

SELECT * FROM users
WHERE user_password IS NOT NULL
```

 * postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
3 rows affected.

user_id	user_first_name	user_last_name	user_email_id	user_email_validated
3	Gordan	Bradock	gbradock0@barnesandnoble.com	False
4	Tobe	Lyness	tlyness1@paginegialle.it	False
5	Addie	Mesias	amesias2@twitpic.com	False