

# Useful Queries

## Query to find number of devices per device attribute

```
SELECT "device_ip", "device_active", protocols, deviceattributes
[*'attr_type']* , deviceattributes[1]['attr_type'] as type,deviceattributes[1]
['attr_name']::string as attrname, "deviceattributes", "attribute_type"

FROM "devicemodel" where deviceattributes[1]['attr_name']::string = :attr_name
(String,')
```

## Query to show a specific attribute for a host

What if customer wants to see sysObjectID attribute value for a host? In general, attributes are defined in deviceattributes array (field in devicemodel and devicemodel\_allzones views)

Each element of the array looks like:

```
{
  • device_id: 37,
  • attr_type: "ASName",
  • attr_name: "LUMETA",
  • userdefined: false
}
```

so if we want to find information about attribute sysObjectID, we need to filter the field to return us attr\_type set to sysObjectID.

- filter(deviceattributes, [x] -> x['attr\_type'] = 'sysObjectID')['attr\_name\*'] should get you attribute name for sysObjectID.

Here is an example in the query as below:

```
select filter(deviceattributes, [x] -> x['attr_type'] = 'sysObjectID')
[*'attr_name']*] as attributename, device_ip from devicemodel where
device_ip='65.246.245.25'
```

## Quick search for tables by schema, specifying certain tables in that schema

```
observer=#\dt system.*: contains zone, user, collector

observer=#\dt events.*: contains event, notification

observer=#\dt zone.* contains target, attribute, device, interface, etc

observer=#\dt zone_000x.* information of particular zone. Note the 'x' will
be the id in select * from system.zone

observer=#\dn to list schemas
```

## Show distinct IPs on system: good for exceeding license problems

```
select count(distinct ip) from zone.device;

select zone_id, count(distinct ip) from zone.device GROUP BY zone_id;
```

## Query to show all collector-related events

```
select id, type, time, details from event.notification_all where type like
'COLLEC%';
```

## Query to show all collector-related events initiated by user admin

```
select id, type, time, details from event.notification_all where type like
'COLLEC%' and details && array['user:admin'::text];
```

### Quick jump to query . . .

[Find number of devices per device attribute.](#)

[Show me specific attribute value for a given IP](#)

[Search for tables by schema, specifying certain tables in that schema.](#)

[Show distinct IPs on system: good for exceeding license problems.](#)

[Show all collector-related events.](#)

[Show all collector-related events initiated by user admin.](#)

[Show all collector-related events filtered by specific date.](#)

[Show all collector-related events filtered by specific date initiated by user admin.](#)

[Show all active db queries.](#)

[Reset the age-out value for a device.](#)

## Query to show all collector-related events filtered by specific date

```
select id, type, time, details from event.notification_all where type like 'COLLEC%' and time > to_timestamp ('201-0-08 00:00:00', 'YYYY-MM-DD HH24:MI:SS');
```

## Query to show all collector-related events filtered by specific date initiated by user admin

```
select id, type, time, details from event.notification_all where type='COLLECTOR_UPDATED' and time > to_timestamp ('201-0-08 00:00:00', 'YYYY-MM-DD HH24:MI:SS') and details && array['user:admin'::text];
```

## Query to show all active db queries

```
select * from pg_stat_activity where state='active';
```

## Query to reset the age-out value for a device

```
update system.config set value='1' where key='maxDeviceAge';
```

where the value is in number of days. This query will reset your device age out time from 7 days (which is by default) to 1 day (that the customer is setting).