



# Snowflake Listing Technical Guide

A brief guide to getting started with IPinfo's database products on Snowflake.

This article explains how to get up and running with your IPinfo database. We cover the basic aspects of using our database products on the Snowflake platform and summarize a few key concepts of Snowflake.

If you need any further assistance, please do not hesitate to [reach out to us](#).

The infographic features a blue background with a white snowflake icon on the left and the IPinfo logo on the right. The title "IPinfo Databases on Snowflake" is centered at the top. Below the title, there are 12 database product boxes arranged in three rows. Each box contains the name of the database and a small icon indicating its status: a star for "Most popular database", a stack of books for "Contains multiple databases", and a padlock for "Open access databases".

Database Name	Icon
IP Address to Geolocation	★
Privacy / VPN Detection	★
IP Address to Company	★
IP Address to Mobile Carrier	
IP Address to ASN	
IP Address to Hosted Domains	
IP Address to rDNS	
IP Address to WHOIS Summary	📖
IP Address to WHOIS Full Data	📖
All-in-one Full Database	📖
IP Address to Country + ASN	🔒

★ Most popular database    📖 Contains multiple databases    🔒 Open access databases

## **Table of Contents**

### Essential Links

[Available listings on the Snowflake Marketplace](#)

[Resources from IPinfo](#)

[Resources from Snowflake](#)

### Database Operations

[IPinfo Basic UDFs \(User Defined Function\)](#)

[IPinfo UDTFs \(User-Defined Table Functions\)](#)

[Views, UDTFs and Database Names](#)

[Accessing our daily updated databases on Snowflake](#)

[Joining our database to your database using UDTFs](#)

[General Sample Queries](#)

[Preview the database](#)

[Looking up an individual IP address](#)

[Converting Start IP & End IP to CIDR/Network Range equivalent](#)

[Describe the database](#)

[Generating random IP addresses](#)

### Database Specific Operation

[IP to Geolocation: Nearest IP addresses](#)

[IP to Geolocation: Get Continent](#)

[IP to ASN: Number of ASNs per country](#)

[Hosted Domains: Lookup the hosted domain data of an IP address](#)

[Hosted Domains: Lookup a domain](#)

### IPinfo Community

### Additional Support

# Essential Links

Before we get started, we recommend you get acquainted with our services, products, snowflake listings and available resources and documentation.

## Available listings on the Snowflake Marketplace

We currently have the following listings available in the Snowflake Marketplace. You can check our databases from [our storefront](#) as well. Each listing page contains the description of the product, available column/field names, use cases, sample queries and other information.

Listing	Description
<a href="#">IP to Geolocation</a>	Get geolocation data from IP addresses.
<a href="#">IP to Privacy Detection</a>	Find out the IP addresses that are associated with VPNs, Proxies, Tor, Apple Relay, and hosting/data center/cloud providers.
<a href="#">IP to Company</a>	Extract company information associated with an IP address.
<a href="#">IP to Mobile Carrier</a>	Detect mobile carrier name, country and other information from IP addresses.
<a href="#">IP to Hosted Domains</a>	Get a list of domains hosted on an IP address.
<a href="#">IP to rDNS</a>	This database provides the domain name associated with an IP address.
<a href="#">IP to ASN</a>	Get ASN and related data from IP addresses.
<a href="#">IP WHOIS Summary</a>	Structured and uniform WHOIS data from all RIRs. Contains 2 Databases.
<a href="#">IP WHOIS – Full dataset</a>	Contains 5 databases that provide the full WHOIS database in a structured and consistent database format.
<a href="#">All in One IP Address Database</a>	A comprehensive collection of every database offering we have. Including our extended databases.
<a href="#">Personalized IP Database</a>	Database listing for requesting customized data solutions.
<a href="#">Country + ASN IP Address Data *</a>	Open access full accuracy IP address database that provides IP to country-level information and ASN details.

## Resources from IPinfo

To support and enable our customers, we regularly publish [articles and resources](#) that focus on using our products on the Snowflake platform.

- [Introducing a simpler way to get IP Address Data in Snowflake](#)
- [How customers use IPinfo's data in Snowflake](#)
- [What makes IPinfo's Snowflake integration ideal for data engineers?](#)
- [IPinfo Recognized as A Leader in Snowflake's Modern Marketing Data Stack](#)
- [Snowflake and IPinfo threat intelligence webinar](#)
- [6 ways users optimize IP data to improve use cases](#)

Also, please feel free to check our [documentation](#).

## Resources from Snowflake

Snowflake has published many articles and resources to enable customers of IPinfo to effectively and efficiently achieve their data goals.

- [Marketplace Monetization — Self service access to Premium datasets](#)
- [\[Video\] DEMO: Python On Snowflake | Snowpark](#)
- [\[Video\] Snowflake Marketplace Demo | Try Before You Buy Experience](#)

Additionally, you should also check out the Snowflake documentation:

- [Snowflake Documentation](#)
- [Snowflake Marketplace Documentation](#)

# Database Operations

Let us discuss some of the basic IPinfo database operations. For this guide, we expect that you are familiar with the Snowflake platform. However, if you need any assistance, [our data experts are happy to guide you](#).

## IPinfo Basic UDFs (User Defined Function)

To ease the process of using our database products and enable our users with the best-performing solutions, we have developed many handy UDFs (User Defined Functions).

 [Snowflake Documentation on UDF](#).

Our basic UDFs are scalar in nature and output one value or multiple values contained within an array.

UDF	Description
<i><b>TO_INT</b></i>	Converts IPv4 to Integer and IPv6 to Hexadecimal equivalent values.
<i><b>TO_IP</b></i>	Convert IPv4 integer value to its equivalent IPv4 IP address string.
<i><b>TO_IP6</b></i>	Convert IPv6 hex values to their IP address format.
<i><b>RANGE_TO_CIDR</b></i>	Converts start_ip and end_ip columns to their network range / CIDR equivalent. Data outputted as an array.
<i><b>TO_JOIN_KEY</b></i>	Creates JOIN_KEY values for IP addresses. Used in the UDTFs.
<i><b>RAND_IP</b></i>	Returns an array of random IPv4 or IPv6 addresses.
<i><b>IP_CONTINENT</b></i>	Returns the continent name for a single IP address lookup.

Additionally, we provide a **COUNTRIES** complimentary database that includes the country codes, names of countries and the name of the continent.

## IPinfo UDTFs (User-Defined Table Functions)

The UDTF or Table Function returns tabular data. To enable faster and more efficient IP address lookup from our database, we have developed these UDTFs. These custom functions remove the guesswork from querying our database and ensure best practices.

UDTF	Database
<i>IP_LOCATION</i>	IP to Geolocation Database.
<i>IP_PRIVACY</i>	IP to Privacy Detection Database.
<i>IP_COMPANY</i>	IP to Company Database.
<i>IP_CARRIER</i>	IP to Mobile Carrier Database.
<i>IP_ASN</i>	IP to ASN Database.
<i>IP_HOSTED_DOMAINS</i>	IP to Hosted Domains Database.
<i>IP_RIR_WHOIS</i>	IP to RIR WHOIS Database.
<i>IP_RWHOIS</i>	IP to RWHOIS Database.
<i>IP_WHOIS_NET</i>	IP to WHOIS NET Database.
<i>IP_DETAILS</i>	IP to Country + ASN database. Returns country and ASN details.
<i>IP_COUNTRY_DETAILS</i>	IP to Country + ASN database. Returns country and continent location details.
<i>IP_ASN_DETAILS</i>	IP to Country + ASN database. Returns ASN details.
<i>IP_LOCATION_EXTENDED</i>	IP to Geolocation Extended database.
<i>IP_PRIVACY_EXTENDED</i>	IP to Privacy Detection Extended database.

The UDTFs are available in the **public** schema of the database listing. When you purchase our listing, you can optionally provide a name to the shared database. You can learn about this from our [IPinfo community post](#).

For example:

if you have purchased our [IP geolocation listing](#) and named our shared database as “**ipinfo**”, then your functions, views and tables will be available in “**ipinfo.public**” namespace where “**public**” is the name of the shared database and “**public**” is called the schema.

You can access the *IP\_LOCATION* UDTF by calling “**ipinfo.public.ip\_location**”, and access the *IP\_GEOLOCATION* view by calling “**ipinfo.public.ip\_geolocation**”.

If you have access to the database product, you will have access to the associated UDTF. Each UDTF corresponds to its respective database.

## Views, UDTFs and Database Names

Reference list for the database names, views and UDTFs.

**Database Name:** The database name that gets assigned when you purchase or trial our database listing. We recommend you change the name upon purchasing the database, as the standard database names can be quite long. We are mentioning the standard database names here for only reference's sake.

**View:** This is where our data lives. Please note the composite database listings may contain multiple views as they usual contain multiple databases.

**UDTF:** Table function that is used to query our IP databases.

**Composite listings:** Some of our listings contains multiple IP databases. They contain multiple views and multiple UDTFs under one database. For example:

- [IPinfo All-in-One IP Address Database](#)
- [IPinfo Personalized IP Address Database](#) (Our custom database listing)
- [IPinfo Whois Full Dataset](#)
- [IPinfo Whois Summary Dataset \(RIR\)](#) (Contains one view, but 3 UDTFs)

Listings	Standard Listing Name	View	UDTF
<a href="#">IPinfo Geolocation IP Address Data</a>	IPINFO_GEOLOCATION_IP_ADDRESS_DATA	IP_GEOLOCATION	IP_LOCATION
<a href="#">IPinfo Company IP Address Data</a>	IPINFO_COMPANY_IP_ADDRESSES_DATA	IP_COMPANY	IP_COMPANY
<a href="#">IPinfo All-in-One IP Address Database</a>	IPINFO_ALLINONE_IP_ADDRESSES_DATABASE	<contains all the available views>	<contains all the available UDTFs>
<a href="#">IPinfo Personalized IP Address Database</a>	IPINFO_PERSONALIZED_IP_ADDRESS_DATABASE	<custom database>	<custom database>
<a href="#">IPinfo IP Address Data for Privacy Detection (VPN, TOR, Relay, Proxy etc.)</a>	IPINFO_IP_ADDRESS_DATA_FOR_PRIVACY_DETECTION_VPN_TOR_RELAY_PROXY_ETC_	IP_PRIVACY	IP_PRIVACY
<a href="#">IPinfo ASN Data</a>	IPINFO_ASN_DATA	IP_ASN	IP_ASN

Listings	Standard Listing Name	View	UDTF
<a href="#">IPinfo Whois Full Dataset</a>	IPINFO_WHOIS_FULL_DATASET	IP_WHOIS_NET	IP_WHOIS_NET
		IP_WHOIS_ASN	-
		IP_WHOIS_MNT	-
		IP_WHOIS_ORG	-
		IP_WHOIS_POC	-
<a href="#">IPinfo Whois Summary Dataset (RIR)</a>	IPINFO_WHOIS_SUMMARY_DATASET_RIR	IP_RIR_WHOIS	IP_RIR_WHOIS
		IP_RWHOIS	IP_RWHOIS
<a href="#">IPinfo IP Address Data for Mobile Carrier Detection</a>	IPINFO_IP_ADDRESS_DATA_FOR_MOBILE_CARRIER_DETECTION	IP_CARRIER	IP_CARRIER
<a href="#">IPinfo Hosted Domains Data</a>	IPINFO_HOSTED_DOMAINS_DATA	HOSTED_DOMAINS	IP_HOSTED_DOMAINS
<a href="#">IP to rDNS (Reverse DNS Resolution)</a>	IP_TO_RDNS_REVERSE_DNS_RESOLUTION	IP_RDNS_DOMAINS	IP_RDNS_DOMAINS
<a href="#">IPinfo Country + ASN IP Address Data</a>	IPINFO_COUNTRY_ASN_IP_ADDRESS_DATA	IP_COUNTRY_ASN	IP_DETAILS
			IP_COUNTRY_DETAILS
			IP_ASN_DETAILS

Note that, for IP to Geolocation, IP to Hosted Domains and IP to Country and ASN databases the names of the views are not the same as the names of their UDTF.

These extended IP address databases are available through the [personalized IP address database listing](#). These databases provide additional fields for context and accuracy.

Listings	View	UDTF
IPinfo IP to Geolocation Extended Database	IP_LOCATION_EXTENDED	IP_LOCATION_EXTENDED
IPinfo IP to Privacy Extended Database	IP_PRIVACY_EXTENDED	IP_PRIVACY_EXTENDED

## Accessing our daily updated databases on Snowflake

Our standard database listings on Snowflake is updated on a monthly basis. However, we offer a daily updated database as part of our [personalized listing](#) and [All in One listing](#). The standard table, views and UDTFs are available for the daily updated listings, but the underlying database is, as expected going to be the daily updated database. Learn more from our [IPinfo community post](#).



## Joining our database to your database using UDTFs

Our UDTFs provide a highly efficient way of joining our database insights to your log or column of IP addresses. We have listed all the UDTFs available in the previous section.

Say, we have the **logs** database that looks something like this:

IP
106.133.126.58
...
104.254.39.96

**IP\_LOCATION** UDTF using the IP to Geolocation database:

```
SELECT *  
FROM logs l  
JOIN TABLE(ipinfo.public.ip_location(l.ip))
```

### Result

IP	CITY	REGION	COUNTRY	...	TIMEZONE
<b>42.77.199.195</b>	Kaohsiung	Takao	TW	...	Asia/Taipei
<b>106.79.74.115</b>	Pune	Maharashtra	IN	...	Asia/Kolkata
...	...	...	...	...	...
<b>104.254.39.96</b>	Oak Hall	Virginia	US	...	America/New_York
<b>106.133.126.58</b>	Osaka	Ōsaka	JP	...	Asia/Tokyo

### UDTF Code Template

```
SELECT <column_containing_ip_addresses>  
FROM <input_table> <table_alias>  
JOIN TABLE(ipinfo.public.<UDTF_name>(<table_alias>.<column_name>))
```

```

1088
1089     SELECT *
1090     FROM log l
1091     LEFT JOIN TABLE(ipinfo.public.ip_location(l.ip))
1092

```

Objects Editor Results Chart

	IP	CITY	REGION	COUNTRY	POSTAL	LAT	LNG	TIMEZONE
1	172.70.134.241	Ashburn	Virginia	US	20147	39.04372	-77.48749	America/New_York
2	93.40.195.0	Taranto	Apulia	IT	74122	40.4644	17.2471	Europe/Rome
3	162.158.74.142	Manchester	England	GB	M11	53.479	-2.1818	Europe/London
4	166.137.97.0	Pascagoula	Mississippi	US	39567	30.36576	-88.55613	America/Chicago
5	2401:4900:2674:29b8:dad5:7c61:f26f:a2e1	Chennai	Tamil Nadu	IN	600001	13.08784	80.27847	Asia/Kolkata
6	147.182.241.41	Santa Clara	California	US	95054	37.3924	-121.9623	America/Los_Angeles
7	2607:fb90:7bc0::	New York City	New York	US	10004	40.71427	-74.00597	America/New_York

### IP\_HOSTED\_DOMAINS UDTF uses the IP to Hosted Domains Database

```

WITH sample_ips (ip) AS (
    SELECT * FROM
    (VALUES ( '8.8.8.8' ),
    ( '198.35.26.98' )
    )
)

SELECT * FROM (
    SELECT ip FROM sample_ips -- sample log database
) logs JOIN TABLE(ipinfo.public.ip_hosted_domains(logs.ip));

```

IP	TOTAL	DOMAINS
<b>8.8.8.8</b>	13,886	[ "aonode.com", "000180.top", "bits-hyderabad.ac.in", "itempurl.com", ...
<b>198.35.26.98</b>	102	[ "wikipedia.com", "wikidpedia.org", "wikimedia.is", "wikiquote.com", ...

With our open access **IPinfo Country + ASN IP Address Data** database, we provide three different UDTFs:

1. **IP\_DETAILS:** Provides all the information available (country, continent and AS details) on the listing for the provided IP addresses.

2. **IP\_COUNTRY\_DETAILS**: Returns the location information available (code and name of the country and continent) on the listing for input IP addresses.
3. **IP\_ASN\_DETAILS**: Returns the AS (Autonomous System) details (ASN, AS Name, AS Domain/Website) for provided IP addresses.

```
SELECT *
FROM logs l
JOIN TABLE(ipinfo.public.ip_details(l.ip))
```

```
SELECT *
FROM logs l
JOIN TABLE(ipinfo.public.ip_coutry_details(l.ip))
```

```
1108
1109
1110
1111
1112
```

```
SELECT *
FROM log l
LEFT JOIN TABLE(ipinfo.public.ip_details(l.ip))
```

Objects Editor Results Chart

	IP	COUNTRY	COUNTRY_NAME	CONTINENT	CONTINENT_NAME	ASN	AS_NAME	...	AS_DOMAIN
1	49.126.146.0	NP	Nepal	AS	Asia	AS38565	Ncell Pvt. Ltd.		axiata.com
2	5.117.218.134	IR	Iran	AS	Asia	AS44244	Iran Cell Service and Communication Company		mtnirancell.ir
3	93.40.195.0	IT	Italy	EU	Europe	AS12874	Fastweb SpA		fastweb.it
4	2401:4900:2674:29b8:dad5:7c61:f26f:a2e1	IN	India	AS	Asia	AS45609	Bharti Airtel Ltd. AS for GPRS Service		airtel.com
5	2607:fb90:7bc0::	US	United States	NA	North America	AS21928	T-Mobile USA, Inc.		t-mobile.com
6	128.199.234.169	SG	Singapore	AS	Asia	AS14061	DigitalOcean, LLC		digitalocean.com
7	147.182.241.41	US	United States	NA	North America	AS14061	DigitalOcean, LLC		digitalocean.com
8	162.158.74.142	GB	United Kingdom	EU	Europe	AS13335	Cloudflare, Inc.		cloudflare.com
9	166.137.97.0	US	United States	NA	North America	AS20057	AT&T Mobility LLC		att.com
10	172.70.134.241	US	United States	NA	North America	AS13335	Cloudflare, Inc.		cloudflare.com

There is also 3 different UDTF for WHOIS database query, for example, the **IP\_RIR\_WHOIS UDTF** queries the RIR WHOIS database.

```
WITH sample_ips (ip) AS (
  SELECT * FROM
  (VALUES ( '1.1.1.1' ),
    ( '81.143.28.80' )
  )
)

SELECT * FROM (
  SELECT ip FROM sample_ips -- sample Log database
) logs JOIN TABLE(ipinfo.public.ip_rir_whois(logs.ip));
```

You only need to pass a column that contains the IP addresses and the UDTF function with take care of the rest.

Let's look at another UDTF, **IP\_PRIVACY**, which comes with our IP to Privacy Detection Database. We will be using our logs table again.

```
SELECT *  
FROM log l  
JOIN TABLE(ipinfo.public.ip_privacy(l.ip))
```

**Result**

IP	HOSTING	PROXY	TOR	VPN	RELAY	SERVICE
104.193.114.122				TRUE		

Please note that even though the logs table contains multiple IP addresses, here we can see only 1 row. That is because the code used performed an INNER JOIN.

```
1088  
1089 SELECT *  
1090 FROM log l  
1091 JOIN TABLE(ipinfo.public.ip_privacy(l.ip))  
1092
```

Objects Editor Results Chart

	IP	HOSTING	PROXY	TOR	VPN	RELAY	SERVICE ...
1	147.182.241.41	TRUE	null	null	null	null	null
2	162.158.74.142	TRUE	null	null	null	TRUE	null
3	172.70.134.241	TRUE	null	null	null	TRUE	null

Now, if you have purchased multiple database products from us, or our All-in-one database and want to perform multiple UDTF operations and want to join them into a comprehensive table, you should run an OUTER JOIN operation.

```
SELECT
  l.ip, -- Log IP address column
  carrier.name carrier, -- asn columns
  -- ...
  carrier.mnc mnc,
  privacy.vpn vpn, -- privacy columns
  -- ...
  privacy.service privacy_service,
  ip_geo.city city, -- ip geolocation columns
  -- ...
  ip_geo.timezone timezone
FROM logs l
LEFT JOIN (
  SELECT *
  FROM logs l
  JOIN TABLE(ipinfo.public.ip_carrier(l.ip))
) carrier
ON l.ip = carrier.ip
LEFT JOIN (
  SELECT *
  FROM logs l
  JOIN TABLE(ipinfo.public.ip_privacy(l.ip))
) privacy
on l.ip = privacy.ip
LEFT JOIN (
  SELECT *
  FROM logs l
  JOIN TABLE(ipinfo.public.ip_location(l.ip))
) ip_geo
on l.ip = ip_geo.ip
```

This will result in a comprehensive table that will contain geolocation, privacy and carrier data of the log IP table.

## General Sample Queries

In this section, we will cover some of the general and basic sample queries that can be applied to any database products we have.

### Preview the database

Preview the database by selecting the first 10 rows.

```
SELECT *
FROM ipinfo.public.IP_RIR_WHOIS
LIMIT 10
```

### Looking up an individual IP address

You can look up individual IP addresses using UDTFs as well.

```
-- input IP address: 8.8.8.8
SELECT *
FROM TABLE(ipinfo.public.ip_location('8.8.8.8'))
```

```
-- input IP address: 8.8.8.8
SELECT *
FROM TABLE(ipinfo.public.ip_location('8.8.8.8'))
```

Results Chart

	CITY	REGION	COUNTRY	POSTAL	LAT	LNG	TIMEZONE
1	Mountain View	California	US	94043	37.4056	-122.0775	America/Los_Angeles

### Converting Start IP & End IP to CIDR/Network Range equivalent

You can convert the Start\_IP and End\_IP to their CIDR or network range equivalent using **RANGE\_TO\_CIDR** UDF. Please note that this will result in an array where there will be one or more items.

```
SELECT
  ipinfo.public.range_to_cidr(start_ip, end_ip) as "CIDR/ Network Range",
  *
FROM ipinfo.public.ip_carrier
LIMIT 10
```

CIDR/ Network Range	START_IP_INT	END_IP_INT	...	MNC
["1.1.125.0/24"]	16874752	16875007	...	10
["1.10.130.0/24"]	17465856	17466111	...	1
...	...	...	...	...
["1.10.213.0/24", "1.10.214.0/24"]	17487104	17487615	...	1

```

1142 SELECT
1143     ipinfo.public.range_to_cidr(start_ip, end_ip) as "CIDR/ Network Range",
1144     *
1145 FROM ipinfo.public.carrier
1146 LIMIT 10

```

	CIDR/ Network Range	START_IP_INT	END_IP_INT	JOIN_KEY	START_IP	END_IP	NAME	COUNTRY	MCC	MNC
1	["1.1.125.0/24"]	16874752	16875007	16842752	1.1.125.0	1.1.125.255	NTT docomo	JP	440	10
2	["1.10.130.0/24"]	17465856	17466111	17432576	1.10.130.0	1.10.130.255	AIS	TH	520	1
3	["1.10.202.0/23"]	17484288	17484799	17432576	1.10.202.0	1.10.203.255	AIS	TH	520	1
4	["1.10.208.0/24"]	17485824	17486079	17432576	1.10.208.0	1.10.208.255	AIS	TH	520	1
5	["1.10.214.0/24"]	17487360	17487615	17432576	1.10.214.0	1.10.214.255	AIS	TH	520	1
6	["1.10.220.0/23"]	17488896	17489407	17432576	1.10.220.0	1.10.221.255	AIS	TH	520	1
7	["1.112.98.0/24"]	24142336	24142591	24117248	1.112.98.0	1.112.98.255	SoftBank	JP	440	20
8	["1.120.0.0/16"]	24641536	24707071	24641536	1.120.0.0	1.120.255.255	Telstra	AU	505	1
9	["1.121.0.0/16"]	24707072	24772607	24707072	1.121.0.0	1.121.255.255	Telstra	AU	505	1
10	["1.122.0.0/16"]	24772608	24838143	24772608	1.122.0.0	1.122.255.255	Telstra	AU	505	1

## Describe the database

Use the **DESCRIBE** or **DESC** ([Documentation](#)) command to get the schema and other information of the table.

```
DESCRIBE TABLE ipinfo.public.ip_rir_whois;
```

```
-- . .
1150
1151 DESCRIBE TABLE ipinfo.public.rir_whois;
```

	name	type	kind	null?	default	primary key	unique key	check	expression	comment	policy name
1	RANGE	VARCHAR(16777216)	COLUMN	Y	null	N	N	null	null	null	null
2	ID	VARCHAR(16777216)	COLUMN	Y	null	N	N	null	null	null	null
3	NAME	VARCHAR(16777216)	COLUMN	Y	null	N	N	null	null	null	null
4	COUNTRY	VARCHAR(16777216)	COLUMN	Y	null	N	N	null	null	null	null
5	STATUS	VARCHAR(16777216)	COLUMN	Y	null	N	N	null	null	null	null
6	TECH	VARCHAR(16777216)	COLUMN	Y	null	N	N	null	null	null	null
7	MAINTAINER	VARCHAR(16777216)	COLUMN	Y	null	N	N	null	null	null	null
8	ADMIN	VARCHAR(16777216)	COLUMN	Y	null	N	N	null	null	null	null
9	SOURCE	VARCHAR(16777216)	COLUMN	Y	null	N	N	null	null	null	null
10	WHOIS_DOMAIN	VARCHAR(16777216)	COLUMN	Y	null	N	N	null	null	null	null
11	UPDATED	DATE	COLUMN	Y	null	N	N	null	null	null	null
12	ORG	VARCHAR(16777216)	COLUMN	Y	null	N	N	null	null	null	null
13	RDNS_DOMAIN	VARCHAR(16777216)	COLUMN	Y	null	N	N	null	null	null	null
14	DOMAIN	VARCHAR(16777216)	COLUMN	Y	null	N	N	null	null	null	null
15	GEOLOC	VARCHAR(16777216)	COLUMN	Y	null	N	N	null	null	null	null
16	ORG_ADDRESS	VARCHAR(16777216)	COLUMN	Y	null	N	N	null	null	null	null
17	ASN	VARCHAR(16777216)	COLUMN	Y	null	N	N	null	null	null	null
18	AS_NAME	VARCHAR(16777216)	COLUMN	Y	null	N	N	null	null	null	null

## Generating random IP addresses

For testing purposes, you might need to generate IP addresses. In that case, you can use the included, **RAND\_IP** UDF. The **RAND\_IP** UDF can generate any number of IPv4 or IPv6 addresses.

The **RAND\_IP** UDF takes two parameters:

- Number of IP addresses: It could be any number of IP addresses.
- IP Address type: For IPv4 it is "4", for IPv6 it is "6".

```
-- SELECT IPINFO.PUBLIC.RAND_IP(<number_of_ips>, <type_of_ip>)
SELECT IPINFO.PUBLIC.RAND_IP(3,6)
```

```
IPINFO.PUBLIC.RAND_IP(3,6)
```

```
[ "d043:4884:c648:55e3", "bdce:c19f:5429:e3d9", "c2a4:1049:b6cf:d90b" ]
```

If you would like to generate a table of IPv4 IP addresses, you can use the following command as well:



```

SELECT CONCAT(
  TRUNCATE( UNIFORM(0, 256, RANDOM()) ), '.',
  TRUNCATE( UNIFORM(0, 256, RANDOM()) ), '.',
  TRUNCATE( UNIFORM(0, 256, RANDOM()) ), '.',
  TRUNCATE( UNIFORM(0, 256, RANDOM()) )
) ips_ipv4
FROM TABLE(GENERATOR(rowcount => 10)) --number of rows => 10
ORDER BY 1;

```

IPS_IPV4
102.98.120.178
107.205.67.256
112.228.52.231
147.146.42.66
166.103.185.88

Please note a couple of things:

- When you are generating random IP address it might include some [bogon IP addresses](#) and you will not find information regarding them when you run a UDTF command on them.
- The vast majority of IPv6 addresses are unassigned, so there is no information associated with them. So, randomized IPv6 addresses cannot be effective for testing purposes.

## Database Specific Operation

Now that we have covered basic guidelines and some of the general sample queries, we can take a look at some of the interesting ways you can use our database. The following queries are fun ways to experiment and find inspiration in using our database for data enrichment.

### IP to Geolocation: Nearest IP addresses

Using the **HAVERSINE** ([documentation](#)) function provided by Snowflake, we can get information of IP addresses nearest to a geographic coordinate.

```

-- Pass the input lat, lng as the first two argument
-- to the Haversine function. (e.g. 51.1740, -1.8262)

```

```

SELECT

```

```

HAVERSINE(51.1740, -1.8262, lat, lng) as distance,
start_ip,
end_ip,
city,
region,
country
FROM ipinfo.public.ip_geolocation
ORDER BY 1
LIMIT 10;

```

DISTANCE	START_IP	END_IP	CITY	REGION	COUNTRY
2.956907252	2a00:2381:26a8::	2a00:2381:26a8:ffff:fff:f fff:ffff:ffff	Countess	England	GB
2.966382522	2a02:26f7:c880:70a1: :	2a02:26f7:c880:70a1:fff f:ffff:ffff:ffff	Lark Hill	England	GB
...	...	...	...	...	...
2.966382522	2a02:26f7:c888:70a1: :	2a02:26f7:c888:70a1:fff f:ffff:ffff:ffff	Lark Hill	England	GB

## IP to Geolocation: Get Continent

To only get the continent information of a single IP address, you can use the **IP\_CONTINENT** UDF.

```
SELECT IPINFO.PUBLIC.IP_CONTINENT('8.8.8.8')
```

However, please note that this UDF only takes an IP address and returns the name of the continent by utilizing the **LOCATION** database and the **COUNTRIES** complementary database. If you are already using the **IP\_LOCATION** UDTF or you have to look up multiple IP addresses, you can join the output table to the **COUNTRIES** complementary database.

```
SELECT *
FROM (
  SELECT ip FROM logs -- sample log database
) logs JOIN TABLE(ipinfo.public.ip_geolocation(logs.ip)) l
LEFT JOIN ipinfo.public.countries c
WHERE l.country=c.cc
```

IP	CITY	REGION	COUNTRY	POSTAL	LAT	LNG	TIMEZONE	CC	NAME	CONTINENT	
28.118.204.125	Columbus	Ohio	US		39.969	-83.0114	America/New_York	US	United States	North America	
208.63.165.46	Southaven	Mississippi	US	38671	34.98898	-90.01259	America/Chicago	US	United States	North America	
138.164.254.15	Columbus	Ohio	US		43218	39.969	-83.0114	America/New_York	US	United States	North America
161.208.204.255	Portland	Oregon	US	97207	45.5235	-122.6762	America/Los_Angeles	US	United States	North America	
113.176.126.211	Đông Hà	Quảng Trị	VN	48199	16.81625	107.10031	Asia/Bangkok	VN	Vietnam	Asia	
51.31.243.14	London	England	GB	EC1A	51.50853	-0.12574	Europe/London	GB	United Kingdom	Europe	
34.250.126.102	Dublin	Leinster	IE	D02	53.33306	-6.24889	Europe/Dublin	IE	Ireland	Europe	
102.110.199.234	As Sanad	Gafsa	TN		34.4628	9.26404	Africa/Tunis	TN	Tunisia	Africa	
<b>IP to Location</b>									<b>Countries</b>		

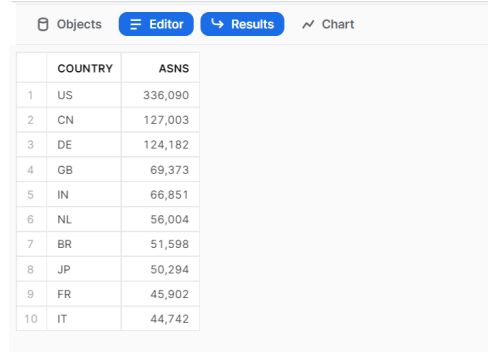
## IP to ASN: Number of ASNs per country

```
SELECT
  country,
  count(1) as ASNs
FROM ipinfo.public.ip_country_asn
GROUP BY 1
ORDER BY 2 DESC
LIMIT 10;
```

```

1136
1137 SELECT
1138     country,
1139     count(1) as ASNs
1140 FROM ipinfo.public.ip_country_asn
1141 GROUP BY 1
1142 ORDER BY 2 DESC
1143 LIMIT 10;

```



	COUNTRY	ASNS
1	US	336,090
2	CN	127,003
3	DE	124,182
4	GB	69,373
5	IN	66,851
6	NL	56,004
7	BR	51,598
8	JP	50,294
9	FR	45,902
10	IT	44,742

## Hosted Domains: Lookup the hosted domain data of an IP address

```

SELECT *
FROM ipinfo.hosted_domains
WHERE ip = '198.35.26.98'

```

IP	TOTAL	DOMAINS
198.35.26.98	102	[ "wikipedia.com", "wikimediafoundation.com", "wikimediacommons.org", ... ]

## Hosted Domains: Lookup a domain

```

SELECT
    *
FROM ipinfo.hosted_domains
WHERE ARRAY_CONTAINS('google.com'::VARIANT, domains);

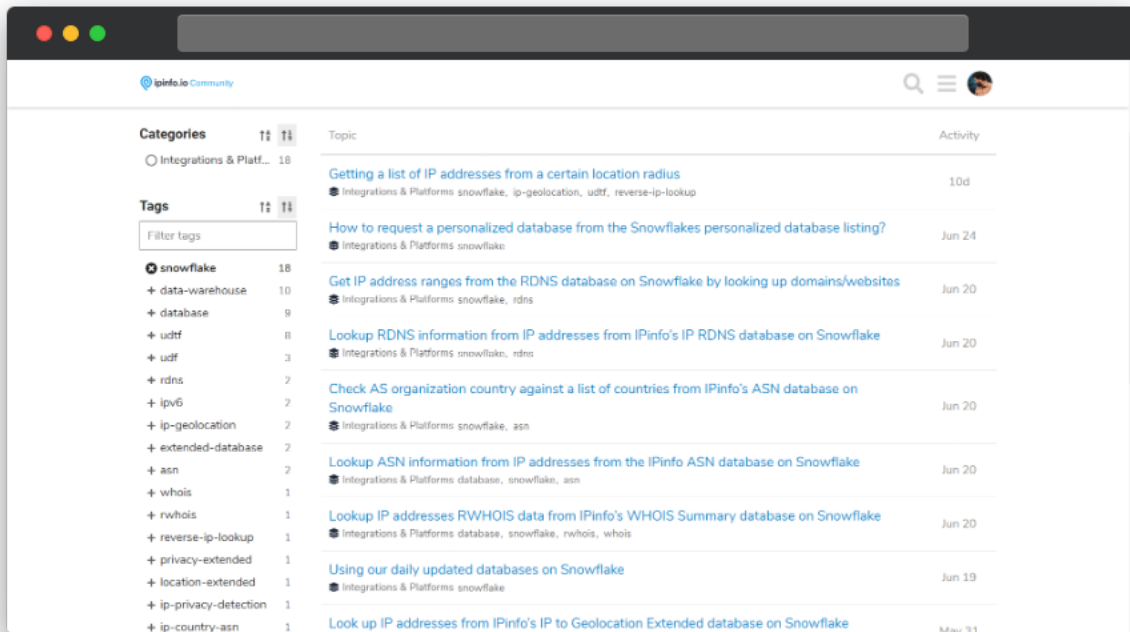
```

If you are looking up domains from the Hosted Domains database frequently, it would be best to [flatten](#) the database before that.

# IPinfo Community

In our [IPinfo community](#), we have created a [repository of all our Snowflake-related queries and technical documentation](#). Through our community, you can learn and explore the technical features of our Snowflake listings as well as ask and participate in discussions about IP data and IPinfo.

[🔗 IPinfo Snowflake Community Documentation](#)



## Additional Support

With the combination of our technical groundwork and our support system, we are one of the leading data providers on the Snowflake Marketplace. In line with our dedication to ensure an absolutely frictionless data journey, we are always keen on listening to you, our customers.

If you have any questions, feedback, guidance or support queries, feel free to reach out to us: [snowflake@ipinfo.io](mailto:snowflake@ipinfo.io)