

[AI & Vectors](#) > [Python Examples](#) > [Developing locally with Vecs](#) >

# Python client

Manage unstructured vector stores in PostgreSQL.

Supabase provides a Python client called `vecs` for managing unstructured vector stores. This client provides a set of useful tools for creating and querying collections in PostgreSQL using the `pgvector` extension.

## Quick start

Let's see how Vecs works using a local database. Make sure you have the Supabase CLI [installed](#) on your machine.

## Initialize your project

Start a local Postgres instance in any folder using the `init` and `start` commands. Make sure you have Docker running!

```
1 # Initialize your project
2 supabase init
3
4 # Start Postgres
5 supabase start
```



## Create a collection

Inside a Python shell, run the following commands to create a new collection called "docs", with 3 dimensions.

```
1 import vecs
2
3 # create vector store client
4 vx = vecs.create_client("postgresql://postgres:postgres@localhost:54322/postgres")
5
6 # create a collection of vectors with 3 dimensions
7 docs = vx.get_or_create_collection(name="docs", dimension=3)
```

## Add embeddings

Now we can insert some embeddings into our "docs" collection using the `upsert()` command:

```
1 import vecs
2
3 # create vector store client
4 docs = vecs.get_or_create_collection(name="docs", dimension=3)
5
6 # a collection of vectors with 3 dimensions
7 vectors=[
8     ("vec0", [0.1, 0.2, 0.3], {"year": 1973}),
9     ("vec1", [0.7, 0.8, 0.9], {"year": 2012})
10 ]
11
12 # insert our vectors
13 docs.upsert(vectors=vectors)
```

## Query the collection

You can now query the collection to retrieve a relevant match:

```
1 import vecs
2
3 docs = vecs.get_or_create_collection(name="docs", dimension=3)
4
5 # query the collection filtering metadata for "year" = 2012
6 docs.query(
7     data=[0.4,0.5,0.6],      # required
8     limit=1,                 # number of records to return
```

```
8     limit=1, # number of records to return
9     filters={"year": {"$eq": 2012}}, # metadata filters
10 )
```

## Deep dive


For a more in-depth guide on `vecs` collections, see [API](#).

## Resources

Official Vecs Documentation: <https://supabase.github.io/vecs/api>

Source Code: <https://github.com/supabase/vecs>

Edit this page on GitHub [↗](#)

 Need some help? [Contact support](#)

 Latest product updates? [See Changelog](#)

 Something's not right? [Check system status](#)

---

© Supabase Inc

Contributing

Author Styleguide

Open Source

SupaSquad

Privacy Settings

