

RAG 示例：人脸向量数据库

- pgvector 介绍文档 <https://cheatsheet.md/zh/vector-database/pgvector.zh>

NODE 环境

```
npm install canvas
--canvas_binary_host_mirror=https://registry.npmmirror.com/-/binary/canvas
```

数据库环境

- <https://pkgs.org/download/postgresql-11-pgvector-dbgsym>
- <https://learn.microsoft.com/zh-cn/azure/cosmos-db/postgresql/howto-use-pgvector>

```
# Add PostgreSQL repository as described on its homepage:#
https://wiki.postgresql.org/wiki/Apt# Update the package index:# Create the
file repository configuration:
sudo sh -c 'echo "deb https://apt.postgresql.org/pub/repos/apt
$(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list' # Import
the repository signing key:
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo
apt-key add -
sudo apt-get update
# Install postgresql-11-pgvector deb package:
apt search postgresql pgvector
sudo apt-get install postgresql-15-pgvector
# 激活插件
su postgres
psql
CREATE EXTENSION IF NOT EXISTS vector
    SCHEMA public;
```

- 修改配置文件 postgresql.conf

```
## 如果插件未在列表中显示，你需要编辑 postgresql.conf 文件并将 vector 插件的名
称添加到 shared_preload_libraries 配置项中。例如：
```

```
shared_preload_libraries = 'vector';
```

请确保你已经正确安装了vector 插件，并且已经在你的PostgreSQL 数据库中启用了该插件。运行以下命令来确认插件是否已加载：

```
su postgres
```

```
psql
```

```
SHOW shared_preload_libraries;
```

保存文件后，重新启动PostgreSQL 服务器以使更改生效。

```
service postgresql restart
```

语句查询

<-> 欧几里得距离

<#> 负内积

<=> 余弦距离

数据范式

- 创建 Schema "FaceReport"
 - user Pointer<_User> 用户指针
 - photo String 类型 正脸照片图片地址
 - feat68 Array 类型 存储面部 68 特征点向量
 - title String 类型 面部数据标题

安装

```
npm i parse-dashboard -g# 启动看板
```

```
parse-dashboard --appId dev --masterKey devmk --serverURL
```

```
http://dev.fmode.cn:1337/parse --appName DevServer
```

- 创建拓展字段

```
-- 数据字段 ALTER TABLE "FaceReport" ADD COLUMN feat68_vector vector(128);
```

测试向量检索语句

```
-- 列出所有的面部数据 SELECT * FROM "FaceReport";
```

```
-- 相似度计算
```

```
select "objectId" as id,"title","photo","createdAt",
```

```
array_agg(item)::vector(128) <=>
```

```
ARRAY[-0.09852483868598938,0.0748210921883583,0.031563639640808105,-0.1
```

```
323895901441574,-0.14048264920711517,-0.012380637228488922,-0.120217472
```

```
3148346,-0.11705487966537476,0.12698683142662048,-0.218950554728508,0.1
```

```
6882860660552979,-0.109050452709198,-0.1343892216682434,-0.032637126743
```

```
79349,-0.08884422481060028,0.238133043050766,-0.1850930005311966,-0.127
```

```
4271309375763, -0.031015701591968536, 0.017501287162303925, 0.082748815417
28973, -0.02009255811572075, -0.03606291860342026, 0.07826575636863708, -0.
07460039108991623, -0.30363327264785767, -0.11924955248832703, -0.01362694
0548419952, -0.045659542083740234, -0.05412667989730835, -0.06456897407770
157, 0.06692294776439667, -0.18031014502048492, -0.0011228621006011963, -0.
009975329041481018, 0.11013024300336838, -0.00493689626455307, -0.08558250
218629837, 0.10225555300712585, 0.02723834663629532, -0.25496453046798706,
0.10651601850986481, 0.05334452539682388, 0.22334304451942444, 0.152255296
70715332, 0.007631596177816391, 0.019638724625110626, -0.1368170976638794,
0.14633668959140778, -0.18137460947036743, 0.026605065912008286, 0.1083736
2706661224, 0.06883929669857025, 0.027700066566467285, 0.04362597316503525,
-0.11393266171216965, 0.050470441579818726, 0.1353151500225067, -0.1138829
8869132996, 0.03125010430812836, 0.10460706800222397, -0.03446819260716438,
-0.01624279096722603, -0.16958579421043396, 0.2160552591085434, 0.17320333
421230316, -0.11017127335071564, -0.25261038541793823, 0.12614592909812927,
-0.096388079226017, -0.08947526663541794, 0.07287181168794632, -0.14514991
641044617, -0.17512711882591248, -0.24459412693977356, 0.02685403823852539,
0.3455549478530884, 0.16986696422100067, -0.16841250658035278, 0.056389730
42368889, -0.006193479523062706, -0.02861897647380829, 0.0987851619720459,
0.18471932411193848, 0.014859762042760849, 0.04832335561513901, -0.0328541
025519371, 0.019863054156303406, 0.2237083464860916, -0.04242844134569168,
-0.017962101846933365, 0.2283974587917328, -0.01908118650317192, 0.0395372
80797958374, -0.044663406908512115, -0.03942383825778961, -0.1094741001725
1968, 0.07384784519672394, -0.13645696640014648, -0.022529728710651398, 0.0
006778687238693237, -0.024799048900604248, -0.04216424375772476, 0.1129275
7093906403, -0.13180139660835266, 0.15981130301952362, -0.0750160589814186
1, 0.037828102707862854, -0.04268859326839447, 0.019408375024795532, -0.098
66605699062347, 0.030573979020118713, 0.11470682919025421, -0.187855228781
70013, 0.1709749549627304, 0.135052889585495, 0.11981010437011719, 0.076126
02412700653, 0.1714332103729248, 0.08336580544710159, 0.03716219961643219,
-0.012700825929641724, -0.23010127246379852, -0.008748907595872879, 0.1327
7511298656464, -0.09321203827857971, 0.08568182587623596, 0.01279112324118
6142]::vector(128) as cos
```

```
from (
  SELECT *, jsonb_array_elements("feat68")::float AS item from (
    SELECT "objectId", "title", "photo", "feat68", "createdAt"
    FROM "FaceReport"
    WHERE "feat68" is not null
    ORDER BY "createdAt" DESC
    limit 20
  ) as rt1
) as rt
GROUP BY "objectId", rt."photo", "createdAt", "title"
ORDER BY cos;
```

