Tidb

2023级　吕欣萌　10222140402　12.17

准备

```
C:\Windows\System32>dir %userprofile%\.ssh\pe-class-key-oregon.pem
 驱动器 C 中的卷是 Windows-SSD
 卷的序列号是 8A8C-F2A6

 C:\Users\86137\.ssh 的目录

2024/12/16  14:32             1,674 pe-class-key-oregon.pem
               1 个文件          1,674 字节
               0 个目录 41,588,342,784 可用字节
```

演示一

步骤一： 在本地打开终端并登录远程 Linux。如果您没有连接凭据，请查看前提准备部分。出现提示时输入回答 yes。

```
C:\Windows\System32>ssh -i %userprofile%\.ssh\pe-class-key-oregon.pem rocky@54.244.137.70
The authenticity of host '54.244.137.70 (54.244.137.70)' can't be established.
ED25519 key fingerprint is SHA256:KAxc2x22Db3MNPprpousqG/vEjdO2okGDTQZyfzJqTM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.244.137.70' (ED25519) to the list of known hosts.
Last login: Mon Jun  3 00:37:29 2024 from 223.104.210.211
```

步骤二：以下操作均在远程 Linux 上进行，使用以下命令连接 TiDB Cloud Serverless。（如需退出 MySQL 客户端时使用 EXIT 命令。）

```
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8842970
Server version: 5.7.28-TiDB-Serverless TiDB Server (Apache License 2.0) Community Edition, MySQL 5.7 compatible

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

步骤三：插入向量和查询向量。

```
mysql> DROP DATABASE IF EXISTS vs_example;
Query OK, 0 rows affected (0.38 sec)

mysql> CREATE DATABASE vs_example;
Query OK, 0 rows affected (0.17 sec)

mysql> USE vs_example;
Database changed
```

```
mysql> SELECT doc, embedding FROM vector_table ORDER BY embedding;
+--------+-----------+
| doc    | embedding |
+--------+-----------+
| apple  | [1,1,1]   |
| banana | [1,1,2]   |
| cat    | [2,2,3]   |
+--------+-----------+
3 rows in set (0.02 sec)
```

步骤四：获取向量字段和特定向量 [1,2,3] 之间的向量余弦距离。

```
mysql> SELECT
    ->   doc, embedding,
    ->   VEC_COSINE_DISTANCE(embedding, '[3,2,1]') AS distance_to_321
    -> FROM vector_table
    -> ORDER BY distance_to_321;
+--------+-----------+--------------------+
| doc    | embedding | distance_to_321    |
+--------+-----------+--------------------+
| apple  | [1,1,1]   | 0.07417990022744858 |
| cat    | [2,2,3]   | 0.15733515938218623 |
| banana | [1,1,2]   | 0.23623738417402662 |
+--------+-----------+--------------------+
3 rows in set (0.02 sec)
```

步骤五：针对 [1,2,3] 在向量列上测试余弦距离、L1 距离（曼哈顿距离）、L2 距离（平方欧几里得距离）和负内积。

```
mysql> SELECT
    ->   embedding,
    ->   VEC_COSINE_DISTANCE(embedding, '[3,2,1]') AS cos_to_321,
    ->   VEC_L1_DISTANCE(embedding, '[3,2,1]') AS l1_to_321,
    ->   VEC_L2_DISTANCE(embedding, '[3,2,1]') AS l2_to_321,
    ->   VEC_NEGATIVE_INNER_PRODUCT(embedding, '[3,2,1]') AS nip_to_321
    -> FROM
    ->   vector_table;
+-----------+--------------------+-----------+-------------------+------------+
| embedding | cos_to_321         | l1_to_321 | l2_to_321         | nip_to_321 |
+-----------+--------------------+-----------+-------------------+------------+
| [1,1,1]   | 0.07417990022744858 |         3 |  2.23606797749979 |         -6 |
| [1,1,2]   | 0.23623738417402662 |         4 | 2.449489742783178 |         -7 |
| [2,2,3]   | 0.15733515938218623 |         3 |  2.23606797749979 |        -13 |
+-----------+--------------------+-----------+-------------------+------------+
3 rows in set (0.01 sec)
```

其他可用的向量函数，如维数、向量与文本之间的转换、欧几里得范数。

```
mysql> SELECT
    ->   embedding,
    ->   VEC_DIMS(embedding) AS "#_of_dim",
    ->   VEC_AS_TEXT(embedding) AS to_text,
    ->   VEC_FROM_TEXT(VEC_AS_TEXT(embedding)) AS embedding_clone,
    ->   VEC_L2_NORM(embedding) AS l2_norm
    -> FROM
    ->   vector_table
    -> ORDER BY embedding;
+-----------+----------+---------+-----------------+--------------------+
| embedding | #_of_dim | to_text | embedding_clone | l2_norm            |
+-----------+----------+---------+-----------------+--------------------+
| [1,1,1]   |        3 | [1,1,1] | [1,1,1]         | 1.7320508075688772 |
| [1,1,2]   |        3 | [1,1,2] | [1,1,2]         | 2.449489742783178  |
| [2,2,3]   |        3 | [2,2,3] | [2,2,3]         | 4.123105625617661  |
+-----------+----------+---------+-----------------+--------------------+
3 rows in set (0.01 sec)
```

演示二

步骤一：以下操作均在实验提供的远程 Linux 上进行，使用以下命令连接 TiDB Cloud Serverless。（要退出 MySQL 客户端可在 MySQL 提示符下使用 EXIT 命令。）

```
mysql: [Warning] Using a password on the command line interface can be insecure.
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8863279
Server version: 5.7.28-TiDB-Serverless TiDB Server (Apache License 2.0) Community Edition, MySQL 5.7 compatible

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

步骤二：将数据（32768 行）填充到示例表 vector_table_noidx 中。

```
mysql> INSERT INTO vector_table_noidx (doc, embedding)
    ->     SELECT NEXTVAL(vs_seq1),
    ->       VEC_FROM_TEXT(CONCAT('[',CONVERT(RAND(), CHAR),',',CONVERT(RAND(), CHAR),',',CONVERT(RAND(), CHAR),']'))
    ->     FROM vector_table_noidx;
Query OK, 8192 rows affected (0.21 sec)
Records: 8192  Duplicates: 0  Warnings: 0

mysql> ANALYZE TABLE vector_table_noidx;

Query OK, 0 rows affected (0.23 sec)

mysql> INSERT INTO vector_table_noidx (doc, embedding)
    ->     SELECT NEXTVAL(vs_seq1),
    ->       VEC_FROM_TEXT(CONCAT('[',CONVERT(RAND(), CHAR),',',CONVERT(RAND(), CHAR),',',CONVERT(RAND(), CHAR),']'))
    ->     FROM vector_table_noidx;
Query OK, 16384 rows affected (0.46 sec)
Records: 16384  Duplicates: 0  Warnings: 0

mysql> ANALYZE TABLE vector_table_noidx;
Query OK, 0 rows affected (0.19 sec)
```

步骤三：查看以下查询。它显示了使用余弦距离的典型 K 最近邻搜索。

```
mysql> EXPLAIN
    -> SELECT doc, embedding
    -> FROM vector_table_noidx
    -> ORDER BY VEC_COSINE_DISTANCE(embedding, '[-1, -1, -1]')
    -> LIMIT 1;
+--------------------------------+---------+-----------+--------------------------+---------------------------------------------------------------------------------------
| id                             | estRows | task      | access object            | operator info
+--------------------------------+---------+-----------+--------------------------+---------------------------------------------------------------------------------------
| Projection_14                  | 1.00    | root      |                          | vs_example.vector_table_noidx.doc, vs_example.vector_table_noidx.embedding
| └─TopN_7                       | 1.00    | root      |                          | Column#4, offset:0, count:1
|   └─Projection_15              | 1.00    | root      |                          | vs_example.vector_table_noidx.doc, vs_example.vector_table_noidx.embedding, vec_cosine_distance(vs_example.vector_table_noidx.e
mbedding, [-1,-1,-1])->Column#4 |
|     └─TableReader_13           | 1.00    | root      |                          | data:TopN_12
|       └─TopN_12                | 1.00    | cop[tikv] |                          | vec_cosine_distance(vs_example.vector_table_noidx.embedding, [-1,-1,-1]), offset:0, count:1
|         └─TableFullScan_11     | 128.00  | cop[tikv] | table:vector_table_noidx | keep order:false
+--------------------------------+---------+-----------+--------------------------+---------------------------------------------------------------------------------------
6 rows in set (0.02 sec)
```

步骤四：创建一个新表 vector_table_with_index_cos，并在列 embedding 上定义 HNSW 索引，并使用表 vector_table_noidx 中的相同数据填充该表。HNSW 索引提供了良好的性能和近似的准确度（在大多数情况下大于 98%）。请注意，创建 HNSW 索引的 COMMENT "HNSW()" 语法将来可能会发生变化。

```
mysql> SELECT *
    -> FROM information_schema.tiflash_replica
    -> WHERE TABLE_SCHEMA = 'vs_example';
+--------------+----------------------------+----------+---------------+-----------------+-----------+----------+
| TABLE_SCHEMA | TABLE_NAME                 | TABLE_ID | REPLICA_COUNT | LOCATION_LABELS | AVAILABLE | PROGRESS |
+--------------+----------------------------+----------+---------------+-----------------+-----------+----------+
| vs_example   | vector_table_with_index_cos |      220 |             2 |                 |         0 |      0.5 |
+--------------+----------------------------+----------+---------------+-----------------+-----------+----------+
1 row in set (0.09 sec)
```

```
mysql> SELECT doc, embedding
    -> FROM vector_table_with_index_cos
    -> ORDER BY VEC_COSINE_DISTANCE(embedding, '[-1, -1, -1]')
    -> LIMIT 1;
+-------+--------------+
| doc   | embedding    |
+-------+--------------+
| dog   | [-1,-2,-1]   |
+-------+--------------+
1 row in set (0.03 sec)
```

```
mysql> EXPLAIN
    -> SELECT doc, embedding
    -> FROM vector_table_with_index_cos
    -> ORDER BY VEC_COSINE_DISTANCE(embedding, '[-1, -1, -1]')
    -> LIMIT 1;
+------------------------------+----------+-----------+----------------------------------+----------------------------------------------------------------------------------------+
| id                           | estRows  | task      | access object                    | operator info                                                                          |
+------------------------------+----------+-----------+----------------------------------+----------------------------------------------------------------------------------------+
| Projection_14                | 1.00     | root      |                                  | vs_example.vector_table_with_index_cos.doc, vs_example.vector_table_with_index_cos.embedding |
| └─TopN_7                     | 1.00     | root      |                                  | Column#4, offset:0, count:1                                                            |
|   └─Projection_15            | 1.00     | root      |                                  | vs_example.vector_table_with_index_cos.doc, vs_example.vector_table_with_index_cos.embedding, vec_cosine_distance(vs_example.vector_table_with_index_cos.embedding, [-1,-1,-1])->Column#4 |
|     └─TableReader_13         | 1.00     | root      |                                  | data:TopN_12                                                                           |
|       └─TopN_12              | 1.00     | cop[tikv] |                                  | vec_cosine_distance(vs_example.vector_table_with_index_cos.embedding, [-1,-1,-1]), offset:0, count:1 |
|         └─TableFullScan_11   | 32768.00 | cop[tikv] | table:vector_table_with_index_cos | keep order:false                                                                      |
+------------------------------+----------+-----------+----------------------------------+----------------------------------------------------------------------------------------+
6 rows in set (0.01 sec)
```

步骤五：除了余弦距离之外，还支持L2距离索引。

```
mysql> SELECT *
    -> FROM information_schema.tiflash_replica
    -> WHERE TABLE_SCHEMA = 'vs_example';
+--------------+-----------------------------+----------+---------------+-----------------+-----------+----------+
| TABLE_SCHEMA | TABLE_NAME                  | TABLE_ID | REPLICA_COUNT | LOCATION_LABELS | AVAILABLE | PROGRESS |
+--------------+-----------------------------+----------+---------------+-----------------+-----------+----------+
| vs_example   | vector_table_with_index_cos |      220 |             2 |                 |         1 |        1 |
| vs_example   | vector_table_with_index_l2  |      223 |             2 |                 |         0 |      0.5 |
+--------------+-----------------------------+----------+---------------+-----------------+-----------+----------+
2 rows in set (0.02 sec)
```

```
mysql> SELECT doc, embedding
    -> FROM vector_table_with_index_l2
    -> ORDER BY VEC_L2_DISTANCE(embedding, '[-1, -1, -1]')
    -> LIMIT 1;
+-------+--------------+
| doc   | embedding    |
+-------+--------------+
| dog   | [-1,-2,-1]   |
+-------+--------------+
1 row in set (0.03 sec)
```

```
mysql> EXPLAIN
    -> SELECT doc, embedding
    -> FROM vector_table_with_index_l2
    -> ORDER BY VEC_L2_DISTANCE(embedding, '[-1, -1, -1]')
    -> LIMIT 1;
+------------------------------+----------+-----------+---------------------------------+----------------------------------------------------------------------------------------+
| id                           | estRows  | task      | access object                   | operator info                                                                          |
+------------------------------+----------+-----------+---------------------------------+----------------------------------------------------------------------------------------+
| Projection_14                | 1.00     | root      |                                 | vs_example.vector_table_with_index_l2.doc, vs_example.vector_table_with_index_l2.embedding |
| └─TopN_7                     | 1.00     | root      |                                 | Column#4, offset:0, count:1                                                            |
|   └─Projection_15            | 1.00     | root      |                                 | vs_example.vector_table_with_index_l2.doc, vs_example.vector_table_with_index_l2.embedding, vec_l2_distance(vs_example.vector_table_with_index_l2.embedding, [-1,-1,-1])->Column#4 |
|     └─TableReader_13         | 1.00     | root      |                                 | data:TopN_12                                                                           |
|       └─TopN_12              | 1.00     | cop[tikv] |                                 | vec_l2_distance(vs_example.vector_table_with_index_l2.embedding, [-1,-1,-1]), offset:0, count:1 |
|         └─TableFullScan_11   | 32768.00 | cop[tikv] | table:vector_table_with_index_l2 | keep order:false                                                                      |
+------------------------------+----------+-----------+---------------------------------+----------------------------------------------------------------------------------------+
6 rows in set (0.01 sec)
```

演示三：

步骤一： 在环境中设置 Zhipu AI API key。将 ${ZHIPUAI_API_KEY} 替换为实际值，并在远程 Linux 上执行以下命令。 在 open.bigmodel.cn 注册后可以获得 tokens 资源包，在页面右上角 API 密钥 模块 可以获得 api_key。

```
[rocky@ip-10-90-1-252 ~]$ export ZHIPUAI_API_KEY=6080...noN7
```

步骤二：安装所需 Python 模块。

```
Successfully installed PyMySQL-1.1.0 annotated-types-0.7.0 anyio-4.7.0 cachetools-5.5.0 certifi-2024.12.14 click-8.1.7 e
xceptiongroup-1.2.2 h11-0.14.0 httpcore-1.0.7 httpx-0.28.1 pydantic-2.10.3 pydantic-core-2.27.1 pyjwt-2.8.0 sniffio-1.3.
1 typing-extensions-4.12.2 zhipuai-2.1.3
```

步骤三：将 TiDB 用作向量存储并通过高效的向量索引检索向量，您需要先配置 TiDB 数据库的连接。

```
[rocky@ip-10-90-1-252 ~]$ ls -l config.py
-rw-r--r--. 1 rocky rocky 486 Dec 18 00:36 config.py
```

步骤四： 接下来，需要模拟一个您个人的知识库，可以将一些您认为有意思知识（最好是 LLM 在训练时候还没有发生的事情）先存储到一个文件中，之后的程序会将它们载入到 TiDB 数据库中。

```
[rocky@ip-10-90-1-252 ~]$ ls -l kb.py
-rw-r--r--. 1 rocky rocky 6444 Dec 18 00:39 kb.py
```

步骤五：创建一个原型聊天机器人，代码在 tidb-chat-no-RAG.py 中， 原型聊天机器人并没有使用 RAG 技术， 而是直接调用 LLM 来回答问题。

```
[rocky@ip-10-90-1-252 ~]$ ls -l tidb-chat-no-RAG.py
-rw-r--r--. 1 rocky rocky 743 Dec 18 00:39 tidb-chat-no-RAG.py
```

```
[rocky@ip-10-90-1-252 ~]$ python tidb-chat-no-RAG.py
请您提问:
```

```
[rocky@ip-10-90-1-252 ~]$    截至我的知识更新日期，历次在美国赢得总统大选的人中，年龄最大的是乔·拜登。他在2020年赢得总统大选时，首次就职时年龄为78岁。这是美国历史上就任时年龄最大的总统。
```

步骤六：下面为原型聊天机器人增加 RAG 功能，用来解决机器人无法回答特定领域的实时性提问的问题，代码在 tidb-chat-RAG.py 中，这将是第一代机器人，它在调用 LLM 来回答问题的时候会使用 RAG 技术。

```
[rocky@ip-10-90-1-252 ~]$ ls -l tidb-chat-RAG.py
-rw-r--r--. 1 rocky rocky 3880 Dec 18 00:42 tidb-chat-RAG.py
```

```
[rocky@ip-10-90-1-252 ~]$    根据您提供的内容，历次在美国赢得总统大选的人中，年龄最大的是特朗普。他在2024年赢得总统选举时现年78岁，并且明年1月就职时，他将成为美国历史上最年长的总统。
```

步骤七：连接指定的 TiDB 实例，确认数据库 test 下已创建表 rag_table

```
mysql> USE test;
Database changed
mysql> SHOW TABLES;
+----------------+
| Tables_in_test |
+----------------+
| rag_table      |
+----------------+
1 row in set (0.02 sec)
```

```
mysql> SHOW CREATE TABLE rag_table\G
*************************** 1. row ***************************
       Table: rag_table
Create Table: CREATE TABLE `rag_table` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(512) DEFAULT NULL,
  `doc` text DEFAULT NULL,
  `doc_vec` vector(1024) DEFAULT NULL COMMENT 'HNSW(distance=cosine)'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin
1 row in set (0.01 sec)
```

演示四：

步骤一：连接指定的 TiDB 实例，确认数据库 test 下已创建表 rag_table

```
mysql> DROP DATABASE IF EXISTS chatdb;
 DATABASE chatdb;
EXITQuery OK, 0 rows affected (0.34 sec)

mysql> CREATE DATABASE chatdb;
Query OK, 0 rows affected (0.16 sec)

mysql> EXIT
Bye
```

步骤二：安装所需 Python 模块。

```
Successfully installed Mako-1.3.8 SQLAlchemy-2.0.30 aiohappyeyeballs-2.4.4 aiohttp-3.11.10 aiosignal-1.3.2 alembic-1.14.0 asttokens-3.0.0 async-timeout-4.0.3 backoff-2.2.1 beautifulsoup4-4.12.3 charset-normalizer-3.4.0 colorlog-6.9.0 dataclasses-json-0.6.7 datasets-3.2.0 decorator-5.1.1 dill-0.3.8 distro-1.9.0 dspy-ai-2.4.12 executing-2.1.0 filelock-3.16.1 frozenlist-1.5.0 fsspec-2024.9.0 greenlet-3.1.1 httpx-0.27.2 huggingface-hub-0.27.0 ipython-8.18.1 jedi-0.19.2 joblib-1.3.2 jsonpatch-1.33 jsonpickle-4.0.1 langchain-0.2.17 langchain-community-0.2.0 langchain-core-0.2.43 langchain-text-splitters-0.2.4 langsmith-0.1.147 marshmallow-3.23.1 matplotlib-inline-0.1.7 multidict-6.1.0 multiprocess-0.70.16 mypy-extensions-1.0.0 networkx-3.2.1 numpy-1.26.4 openai-1.27.0 optuna-4.1.0 orjson-3.10.12 packaging-24.2 pandas-2.2.3 parso-0.8.4 pexpect-4.9.0 prompt-toolkit-3.0.48 propcache-0.2.1 ptyprocess-0.7.0 pure-eval-0.2.3 pyarrow-18.1.0 pydantic-2.7.1 pydantic-core-2.18.2 pygments-2.18.0 python-dateutil-2.9.0.post0 pyvis-0.3.1 regex-2024.11.6 requests-2.32.3 requests-toolbelt-1.0.0 soupsieve-2.6 stack-data-0.6.3 structlog-24.4.0 tenacity-8.5.0 tidb-vector-0.0.14 tqdm-4.67.1 traitlets-5.14.3 typing-inspect-0.9.0 tzdata-2024.2 ujson-5.10.0 wcwidth-0.2.13 wikipedia-1.4.0 xxhash-3.5.0 yarl-1.18.3
```

步骤三：在环境中设置 Zhipu AI API key。将 ${ZHIPUAI_API_KEY} 替换为实际值，并在远程 Linux 上执行以下命令。 在 open.bigmodel.cn 注册后可以获得 tokens 资源包，在页面右上角 API 密钥 模块 可以获得 api_key。

```
[rocky@ip-10-90-1-252 ~]$ export ZHIPUAI_API_KEY=6080...noN7
[rocky@ip-10-90-1-252 ~]$
```

步骤四：通过 Wikipedia Loader 查询 TiDB 构建知识图谱。

```
[rocky@ip-10-90-1-252 ~]$ ls -l build-graph.py
-rw-r--r--. 1 rocky rocky 9077 Dec 18 00:56 build-graph.py
```

步骤五：连接 TiDB，查看刚刚构建的简单知识图谱。

```
[rocky@ip-10-90-1-252 ~]$ mysql \
>     --comments -u '2MFosFYDs8p1g3m.RDrFU3jR' -p'gOCBnA24OmzRi6SYve3Ct8bat6JGcIs%3MXQ' \
>     -h gateway01.us-west-2.prod.aws.tidbcloud.com -P 4000 \
>     --ssl-mode=VERIFY_IDENTITY --ssl-ca=/etc/pki/tls/certs/ca-bundle.crt \
>     -D chatdb
mysql: [Warning] Using a password on the command line interface can be insecure.
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8930607
Server version: 5.7.28-TiDB-Serverless TiDB Server (Apache License 2.0) Community Edition, MySQL 5.7 compatible

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
mysql> SHOW CREATE TABLE entities\G
*************************** 1. row ***************************
       Table: entities
Create Table: CREATE TABLE `entities` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(512) DEFAULT NULL,
  `description` text DEFAULT NULL,
  `description_vec` vector(1024) DEFAULT NULL,
  PRIMARY KEY (`id`) /*T![clustered_index] CLUSTERED */,
  VECTOR INDEX `vec_idx_description_vec`((VEC_COSINE_DISTANCE(`description_vec`)))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin
1 row in set (0.02 sec)
```

```
+----+--------------------------+--------------------------------+---------------------------+
| id | name                     | desc_head                      | VEC_DIMS(description_vec) |
+----+--------------------------+--------------------------------+---------------------------+
|  3 | Apache 2.0               | Apache 2.0 is the open-source  |                      1024 |
| 14 | Backup & Restore (BR)    | Backup & Restore (BR) is a dis |                      1024 |
ling
| 12 | Docker                   | Docker can be used to deploy T |                      1024 |
| 15 | Dumpling                 | Dumpling is a data export tool |                      1024 |
|  5 | F1 papers                | F1 papers from Google also pro |                      1024 |
|  4 | Google's Spanner         | Google's Spanner is a source o |                      1024 |
|  2 | PingCAP                  | PingCAP is the primary develop |                      1024 |
|  8 | Raft consensus algorithm | The Raft consensus algorithm i |                      1024 |
|  1 | TiDB                     | TiDB is an open-source NewSQL  |                      1024 |
|    | 11 | TiDB Ansible            | TiDB Ansible is a playbook for |                      1024 |
) | TiDB Data Migration (DM) is a
| 13 | TiDB Data Migration (DM) | TiDB Data Migration (DM) is a  |                      1024 |
|  9 | TiDB Operator            | TiDB Operator is a method of p |                      1024 |
|  7 | TiFlash                  | TiFlash is a columnstore stora |                      1024 |
|  6 | TiKV                     | TiKV is the storage layer of T |                      1024 |
| 10 | TiUP                     | TiUP is a cluster operation an |                      1024 |
------------------------------+--------------   +----+--------------------------+-----------------------

15 rows in set (0.02 sec)
```

```
+----------+--------------------------------------------------+--------------------------+
| src_name | relation_head                                    | tgt_name                 |
+----------+--------------------------------------------------+--------------------------+
| TiDB     | TiDB is primarily developed and supported by PingC | PingCAP                |
| TiDB     | TiDB is licensed under Apache 2.0.               | Apache 2.0               |
| TiDB     | TiDB drew its initial design inspiration from Goog | Google's Spanner       |
| TiDB     | TiDB also drew design inspiration from Google's F1 | F1 papers              |
| TiDB     | TiKV is the storage layer for TiDB.              | TiKV                     |
 as a columnstore alongside TiKV. | TiFlash          | TiDB     | TiDB uses TiFlash as a columnstore alongside TiKV. | TiFlash
| TiDB     | TiDB uses the Raft consensus algorithm to ensure h | Raft consensus algorithm |
deployed using TiDB Operator in Kubern | TiDB O
| TiDB     | TiDB can be deployed using TiDB Operator in Kubern | TiDB Operator          |
| TiDB     | TiUP is a tool for cluster operation and maintenan | TiUP                   |
| TiDB     | TiDB can be deployed using TiDB Ansible, although  | TiDB Ansible           |
| TiDB     | Docker can be used to containerize and deploy TiDB | Docker                 |
| TiDB     | TiDB Data Migration (DM) is used for migrating dat | TiDB Data Migration (DM) |
Restore (BR) is used for backing up and r |    | TiDB     | Backup & Restore (BR) is used for backing up and r | Backup & Restore (BR)
| TiDB     | Dumpling is used for exporting data from TiDB.   | Dumpling                 |
+----------+--------------------------------------------------+--------------------------+
14 rows in set (0.01 sec)
```

步骤六：测试一下基于 Graph RAG 的程序。

```
[rocky@ip-10-90-1-252 ~]$ -rw-r--r--. 1 rocky rocky 4154 Jun 20 15:14 test-graph.py
```

```
[rocky@ip-10-90-1-252 ~]$ Enter your question:
```

问题一：

```
[rocky@ip-10-90-1-252 ~]$  It seems there might be a misunderstanding, as the provided content does not mention "raccoon" or "umbrella." The con
tent provided discusses the Raft consensus algorithm and TiDB, stating that TiDB uses the Raft consensus
```

问题二：

```
[rocky@ip-10-90-1-252 ~]$    TiKV serves as the storage layer for TiDB. This relationship indicates that TiDB relies on TiKV for its storage cap
abilities, allowing TiDB to manage and process data efficiently while providing features like Hybrid
```

问题三：

```
[rocky@ip-10-90-1-252 ~]$    The most relevant tool for migrating data to TiDB is TiDB Data Migration (DM), which is specifically designed for r
eplicating data from MySQL or MariaDB to TiDB.
```