

# Ответ 3

## Что такое векторная база данных (Vector Database) и чем она принципиально отличается от традиционной реляционной базы данных (например, SQL)?

На основе предоставленного учебного материала, вот подробный ответ на ваш вопрос.

### Введение: Проблема поиска и ее решение

Чтобы понять, что такое векторная база данных, сначала нужно понять проблему, которую она решает. Традиционные системы поиска, как в компьютере, ищут по точным ключевым словам. Если вы ищете документ со словом «отчет», система найдет только те файлы, где есть именно это слово. Но если нужный документ называется «ежеквартальный итог», традиционный поиск его пропустит.

Эта проблема приводит к идее **семантического поиска** — поиска по смыслу, а не по точным словам. Для этого необходимо научить компьютер понимать значение данных (текста, изображений) и представлять его в числовом формате. Таким форматом является **векторное представление (embedding)**, а для работы с ним и были созданы векторные базы данных.

### Что такое вектор и как он представляет смысл?

В контексте искусственного интеллекта (AI), **вектор** — это упорядоченный список чисел. Каждое число в этом списке представляет определенную характеристику или «измерение» исходных данных.

Современные AI-модели, такие как **GPT** или **Claude**, преобразуют сложные данные (например, предложение) в вектор, состоящий из сотен или тысяч чисел. Этот процесс называется созданием **векторного представления (embedding)**.

Ключевая идея заключается в том, что **объекты с похожим смыслом будут иметь похожие векторы**. Например, векторы для предложений «Какая сегодня погода?» и «Будет ли сегодня дождь?» будут находиться в многомерном пространстве очень близко друг к другу, в то время как вектор для «Я люблю пиццу» будет от них далеко.

### Что такое векторная база данных (Vector Database)?

Согласно учебному материалу, **векторная база данных (Vector Database)** — это специализированная система, созданная для эффективного хранения, индексирования и поиска огромного количества векторов.

В качестве конкретного примера в материале приводится **Qdrant** — высокопроизводительная open-source векторная база данных.

Для организации данных в Qdrant используются следующие концепции:

- **Коллекция (Collection):** Аналог таблицы в SQL-базе данных. Это именованное хранилище для векторов, которые обычно имеют одинаковую размерность (например, 1536 измерений).
- **Точка (Point):** Аналог строки в таблице. Каждая точка представляет один объект данных и состоит из:
  1. **ID:** Уникальный идентификатор.
  2. **Вектор (Vector):** Числовое представление (embedding) объекта.
  3. **Полезная нагрузка (Payload):** Необязательные метаданные в формате JSON, связанные с вектором (например, название, год, жанр фильма).

### Принципиальное отличие от традиционной реляционной базы данных (SQL)

Основное и принципиальное отличие векторной базы данных от традиционной реляционной (например, SQL) заключается в **типе данных, с которыми они работают, и в способе выполнения запросов**.

Материал выделяет следующие ключевые различия:

#### 1. Тип данных и структура:

- **Традиционная БД (SQL):** Работает со **структурированными данными**, организованными в таблицы, строки и колонки. Например, таблица users с колонками user\_id, name, email.
- **Векторная БД:** Работает с **неструктурированными данными**, представленными в виде векторов (длинных

списков чисел). Сами исходные данные (текст, изображение) могут храниться как метаданные в Payload.

## 2. Тип поиска (запроса):

- **Традиционная БД (SQL):** Выполняет поиск по **точным совпадениям** или логическим условиям. Запросы используют операторы вроде =, >, <, LIKE. Пример: WHERE user\_id = 123 или WHERE price < 300.
- **Векторная БД:** Выполняет поиск по **сходству (similarity)**. Основной тип запроса — «найди N векторов, наиболее близких к заданному вектору запроса». Это и есть семантический поиск.

## 3. Основная технология поиска:

- **Традиционная БД (SQL):** Использует B-деревья и другие индексы для быстрого поиска точных значений в структурированных колонках.
- **Векторная БД:** Использует алгоритмы **приближенного поиска ближайших соседей (Approximate Nearest Neighbor, ANN)**. Один из самых популярных таких алгоритмов, упомянутый в материале, — **HNSW (Hierarchical Navigable Small World)**. Эти алгоритмы позволяют находить «достаточно близкие» векторы в миллионы раз быстрее, чем полный перебор, жертвуя незначительной долей точности ради огромного прироста в скорости.

## Сводная таблица отличий

|   |                       |  |   |  |                    |  |                        |  |   |  |  |  |  |  |                    |  |   |  |  |  |  |  |                       |  |  |  |  |  |  |  |                           |  |   |  |   |  |
|---|-----------------------|--|---|--|--------------------|--|------------------------|--|---|--|--|--|--|--|--------------------|--|---|--|--|--|--|--|-----------------------|--|--|--|--|--|--|--|---------------------------|--|---|--|---|--|
| Характеристика                                  | Традиционная БД (SQL) |  | Векторная БД (например, Qdrant)                                     |  | :---   :---   :--- |  | <b>Основная задача</b> |  |   |  |  |  |  |  |                    |  |   |  |  |  |  |  |                       |  |  |  |  |  |  |  |                           |  |   |  |   |  |
| Хранение и извлечение структурированных данных. |                       |  | Хранение и поиск неструктурированных данных по смысловому сходству. |  |                    |  | <b>Формат данных</b>   |  | Таблицы, строки, колонки (числа, строки, даты). |  | Векторы (списки чисел) и метаданные (Payload). |  |  |  | <b>Тип запроса</b> |  | Поиск по точному совпадению (WHERE id = 123). |  | Поиск по сходству («найди 5 самых похожих»). |  |  |  | <b>Пример запроса</b> |  | SELECT * FROM products WHERE category = 'electronics'; |  | «Найди товары, похожие на 'удобное офисное кресло'». |  |  |  | <b>Базовая технология</b> |  | Индексы для точного поиска (B-деревья). |  | Индексы для приближенного поиска (ANN, например, HNSW). |  |

Таким образом, если традиционная база данных отвечает на вопрос «Что соответствует этому точному критерию?», то векторная база данных отвечает на вопрос «Что похоже на это по смыслу?».