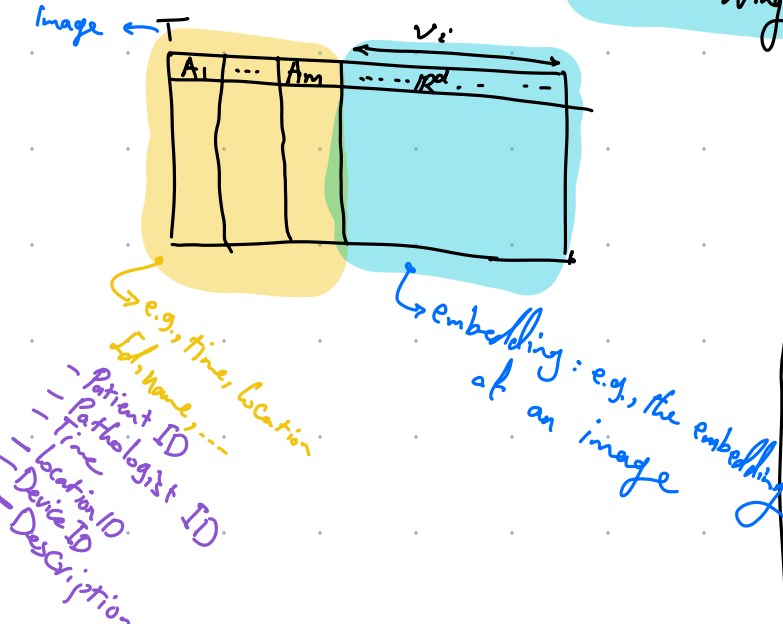


A Relation (Table) in a vector Database is a collection of Tuples (rows) where each row has some **Traditional attributes** and (at least) **one embedding**



Challenges:

- Data Model: Vector Rep. are Machine Understandable while Traditional attributes are Human
- Columns of an embedding are **individually meaningless**
 ↳ Lack of Structure
- Similarities are Vague. we can only measure (approximately) how "similar" to entities are, but we do not know how/why those are similar.
- Incompatibility w/ traditional DBs.
 ↳ different data model, query types
 ↳ different indexing Alg. designed for Sim. Search.
 Curse of
- High-Dimensionalizing
 ↳ Search Space increases exponentially with $d \rightarrow 2^d$

- Non-Constant Comparison
- Traditional attribute $\Theta(1)$
- Embeddings $\Theta(d) \sim [100, 10,000]$

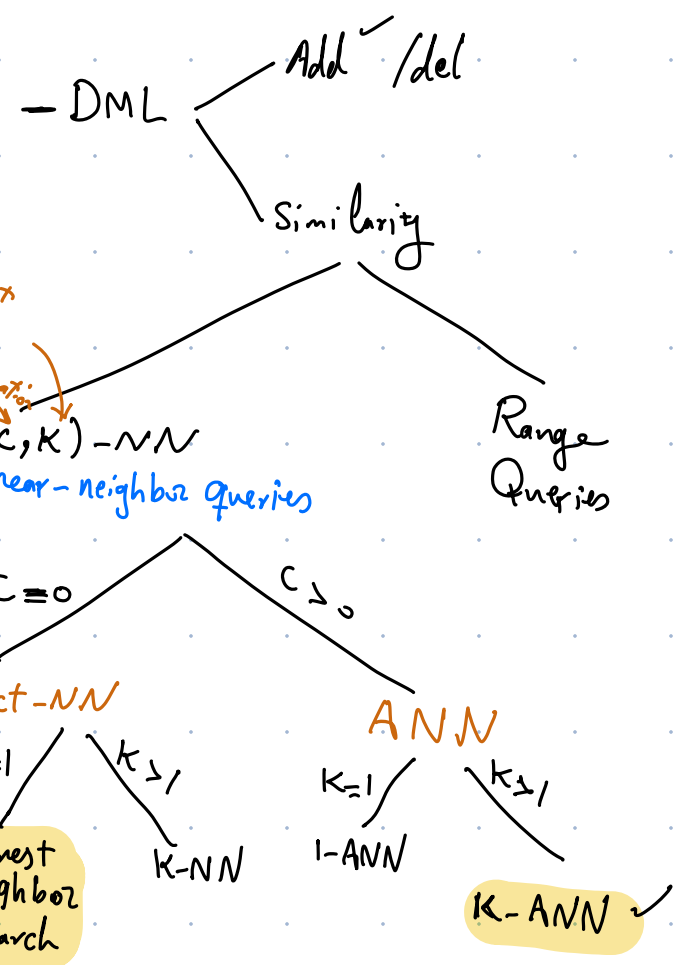
Query Type

Traditional DB

- DDL: Create, drop, ...
 Create/update the Schema
- DML: Select, insert, del, update, ...
 ↳ query/change rows (Tuples)

Vector DB

- DDL: Nothing extra to Traditional DB



Similarity Search

- Single Vector Search
- Constrained Search
 - ↳ Finding similar answers among the ones that satisfy some conditions on traditional attributes.
- Multi-vector Search

Multi → Single Search.

- + every Col. is an embedding
- + query is multiple vectors.

Single → Multi Search.

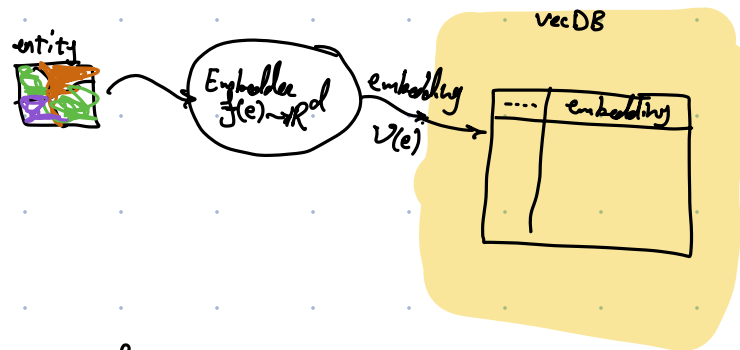
- + every tuple has a SET of embeddings (e.g., documents)
 - ↳ multi-valued attribute
- + the query is a single embedding

- Multi-multi Search.

- + multi-valued embeddings
- + multi-valued query

- Insert/Add : add a new vector representation to DB.

- Direct Insert.



- Benefits:

- easy to adopt
- Flexible for the user.

- Disadvantages:

- Embedding Update

- Semantic Ambiguity

- who knows if the embeddings are compatible

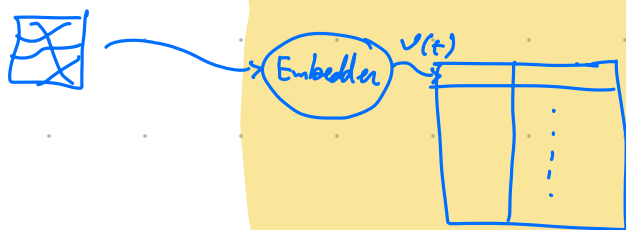
↳ $\delta(t_1, t_2)$ is Legitimate?

* Let $v_1 = f(t) \in \mathbb{R}^d$ and $v_2 = f(t') \in \mathbb{R}^d$
Then v_1 and v_2 are incompatible even though both are in $\mathbb{R}^d$.

In other words, they are in

different embedding spaces

Indirect Input.



Similarity Search queries

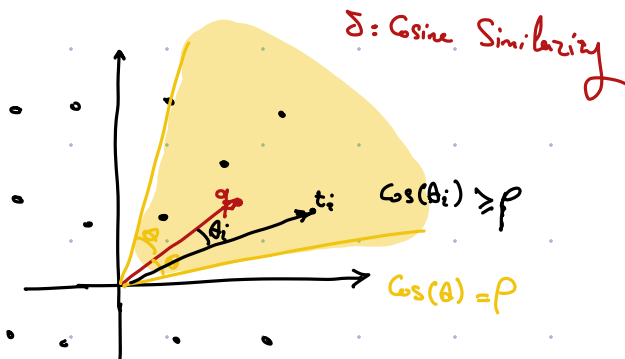
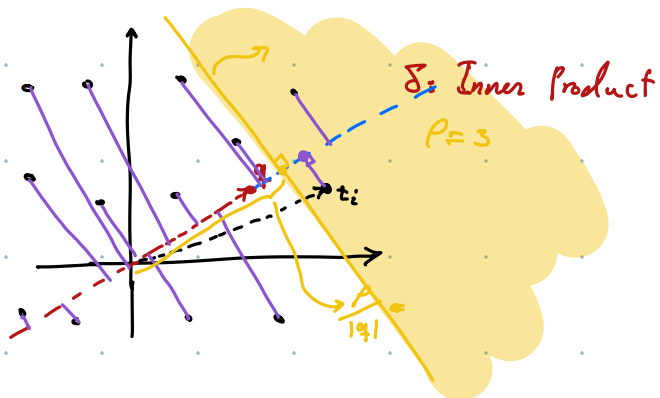
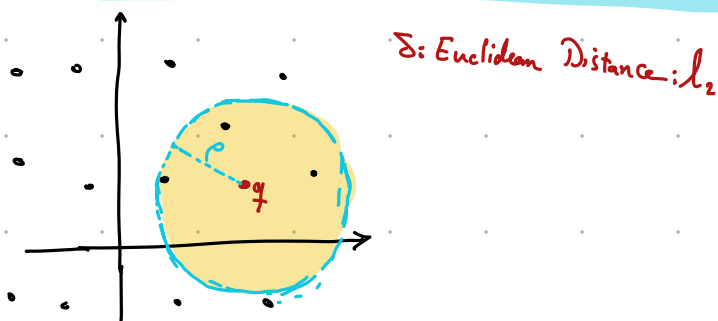
Given a query point $q \in \mathbb{R}^d$

and
 ① radius ρ
 ② a value k

① Range query: Find all tuples that have the distance of at most ρ to q

* $[P_1, P_2]$

$$O = \{t \in D \mid P_1 \leq \delta(v(t), q) \leq P_2\}$$



② K-nearest Neighbors

$$O = \underset{t \in D}{\text{k-argmin}} \delta(v(t), q)$$

(q, k) : kNN of q in D $t \in \mathbb{R}^d$

kNN w/o indexing

Step 1: Compute all distances $\delta(t_i, q) \in \Theta(nd)$

Step 2:

Use the extended median-finding Alg. to return the top-k

$$\Theta(n)$$

$$\Theta(nd)$$

Research Question:

Can we preprocess the data & create a data structure (AKA index) of size $\text{poly}(n, d)$ in time $\text{poly}(n, d)$, such that given a query q , we can answer in $o(n)$