

# **Индекс**

## **Определение**

- Объекты базы данных, предназначенные для ускорения поиска и выборки строк из таблицы, минимизируя количество данных, которые нужно просматривать.
- Так же индексы служат для поддержания некоторых ограничений целостности

# Индекс

## Как работает?

- Устанавливает соответствие между ключом (например, значением проиндексированного столбца) и строками таблицы, в которых этот ключ встречается. Строки идентифицируются с помощью TID (tuple id), который состоит из номера блока файла и позиции строки внутри блока.

# **Индекс**

## **Увеличение затрат на запись**

- При любой операции над проиндексированными данными — будь то вставка, удаление или обновление строк таблицы, — индексы, созданные для этой таблицы, должны быть перестроены, причем в рамках той же транзакции.

# Индекс

## Типы

- B-tree
- Hash
- GIN
- GiST
- BRIN

# Индекс Hash

- Единственная операция, которую поддерживает хеш-индекс, – поиск по условию равенства.
- По мере увеличения количества индексируемых строк одна из корзин расщепляется на две.
- Элементы корзин упорядочены по хеш-кодам ключей, подходящие идентификаторы эффективно находятся двоичным поиском.
- До версии PostgreSQL 10 хеш-индексы не журналировались.

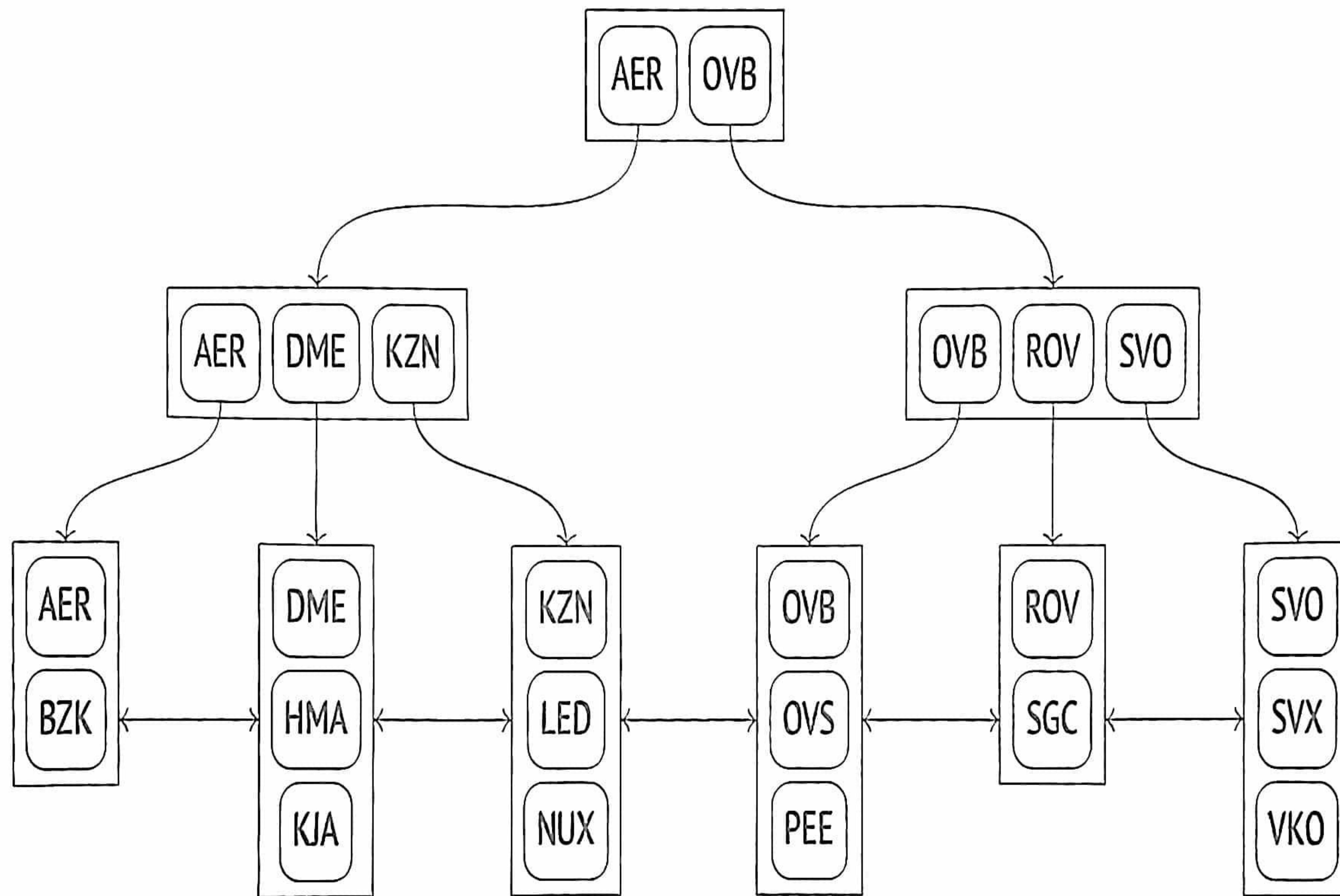
# Индекс

## Hash

- Кластеризация таблицы по хеш-индексу не предусмотрена
- Поскольку хеш-функция не сохраняет отношение порядка, к хеш-индексу неприменимы свойства, касающиеся упорядоченности.
- Хеш-индекс не может участвовать в сканировании только индекса, поскольку не сохраняет ключ индексации и требует перепроверки по таблице.
- Хеш-индекс не работает с неопределенными значениями: операция «равно» не имеет смысла для NULL.
- Поиск значений из массива нереализован.

# Индекс

## B-tree



# Индекс

## B-tree

- Они сбалансированы, то есть все листья находятся на одной глубине. Поэтому поиск любого значения занимает одинаковое время.
- Они сильно ветвисты, то есть каждый узел содержит много элементов, часто сотни. За счет этого глубина B-деревьев получается небольшой даже для очень больших таблиц.
- Данные в индексе упорядочены по возрастанию как между узлами, так и внутри каждого узла. Узлы одного уровня связаны между собой двунаправленным списком.

# Индекс

## B-tree

- Поиск по равенству
- Поиск по неравенству
- Поиск по диапозону
- Поиск по префиксу
- Оптимизация сортировки
- Уникальность

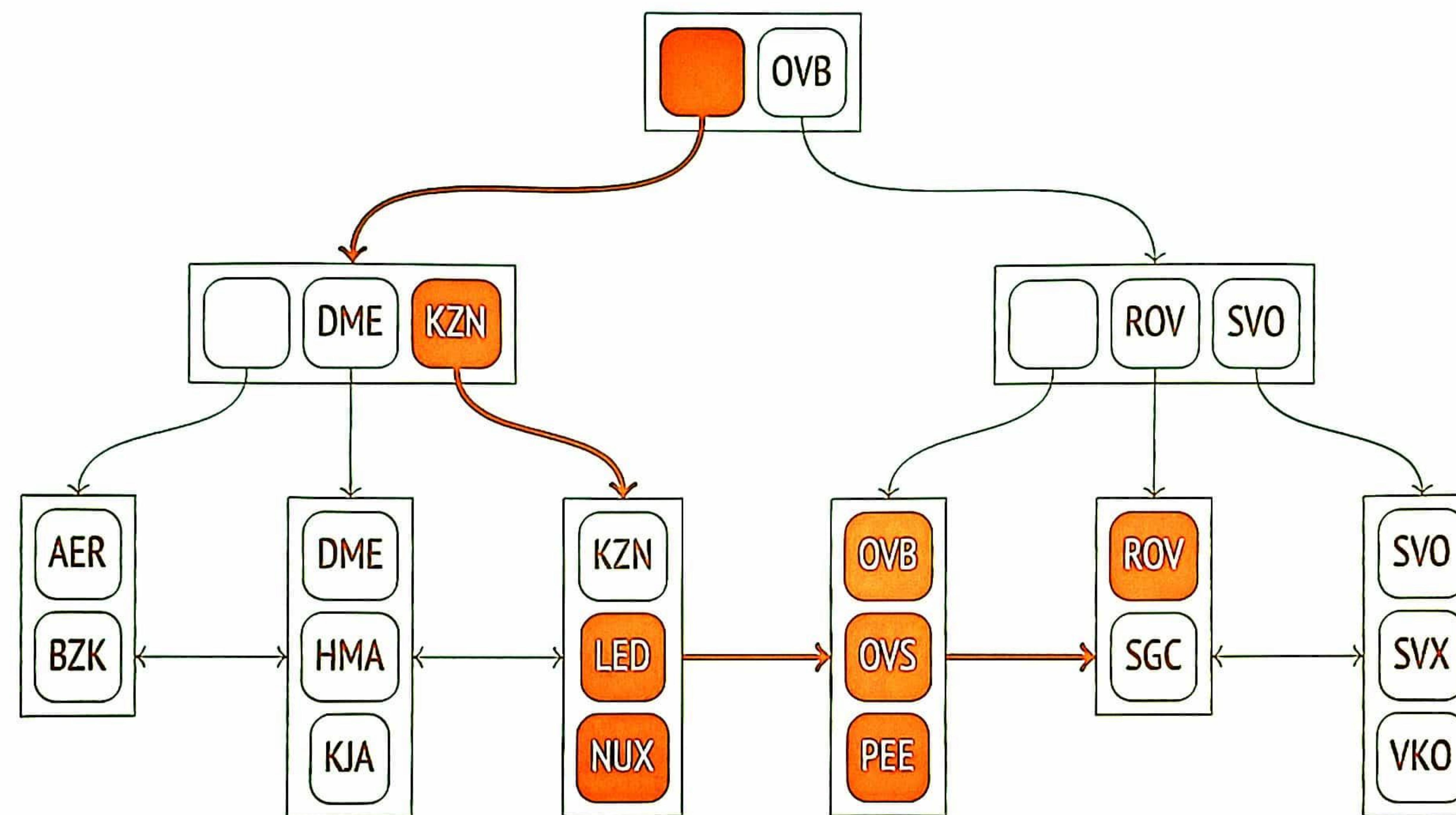
# Индекс

## B-tree

- Дубликаты «схлопываются» в одну индексную запись, содержащую ключ и список табличных идентификаторов.
- Уникальные индексы могут содержать дубликаты ключей из-за многоверсионности, поскольку индекс хранит ссылки на все версии табличных строк.

# Индекс

## B-tree



# **Индекс**

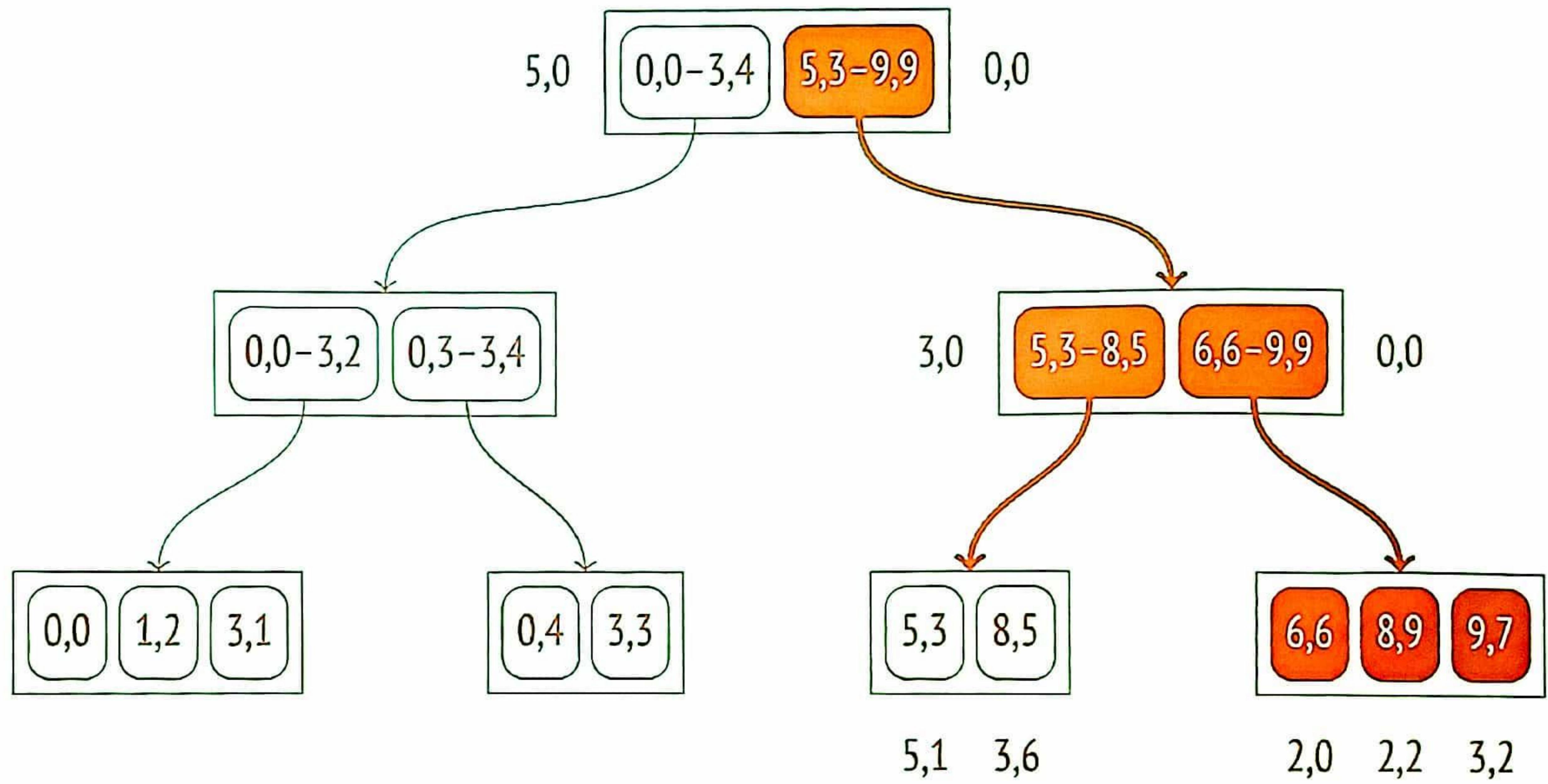
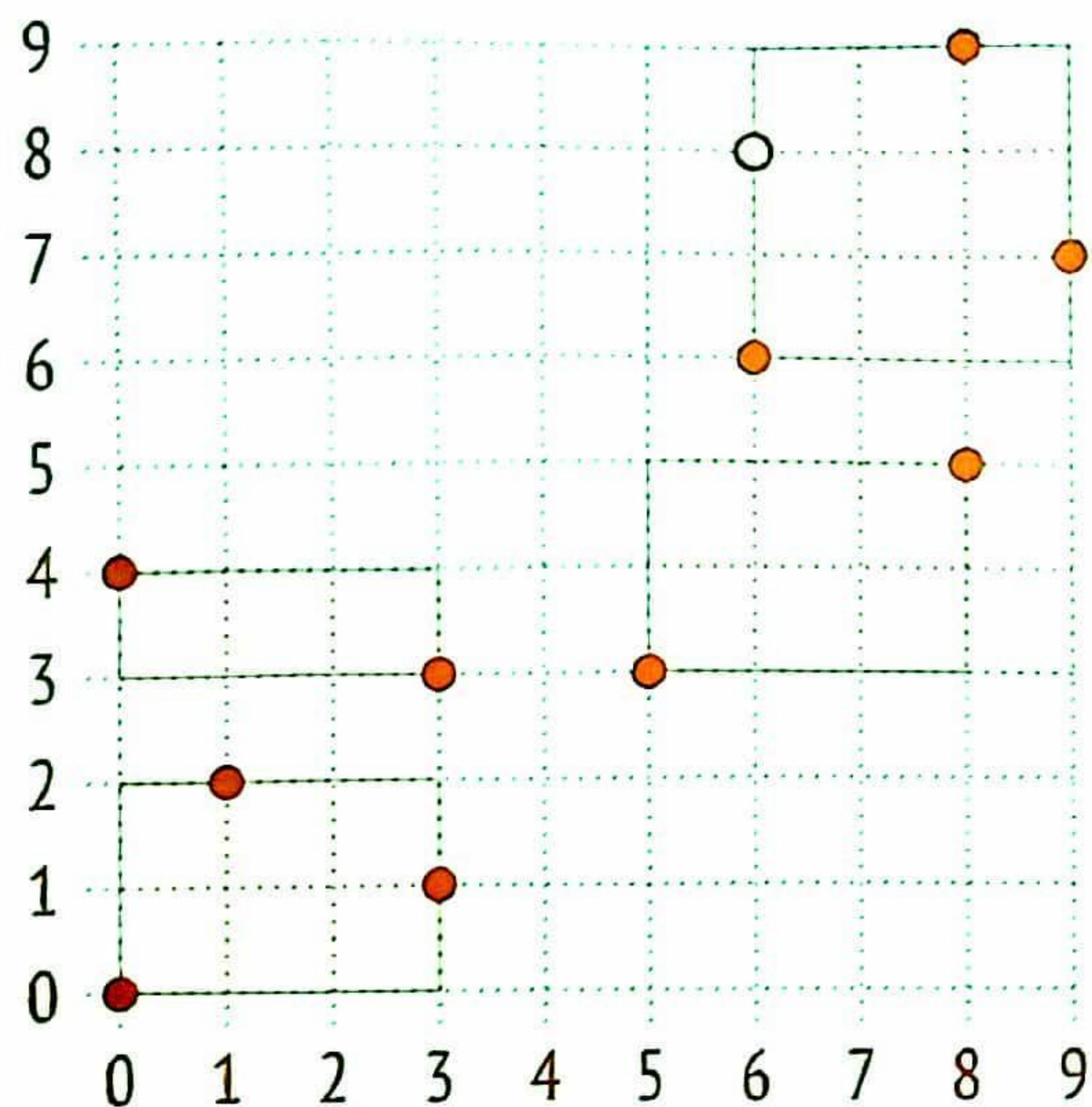
## **GiST (Generalized Search Tree)**

- Не конкретный тип индекса, а фреймворк, который позволяет создавать пользовательские реализации для различных видов данных.

# Индекс R-дерево

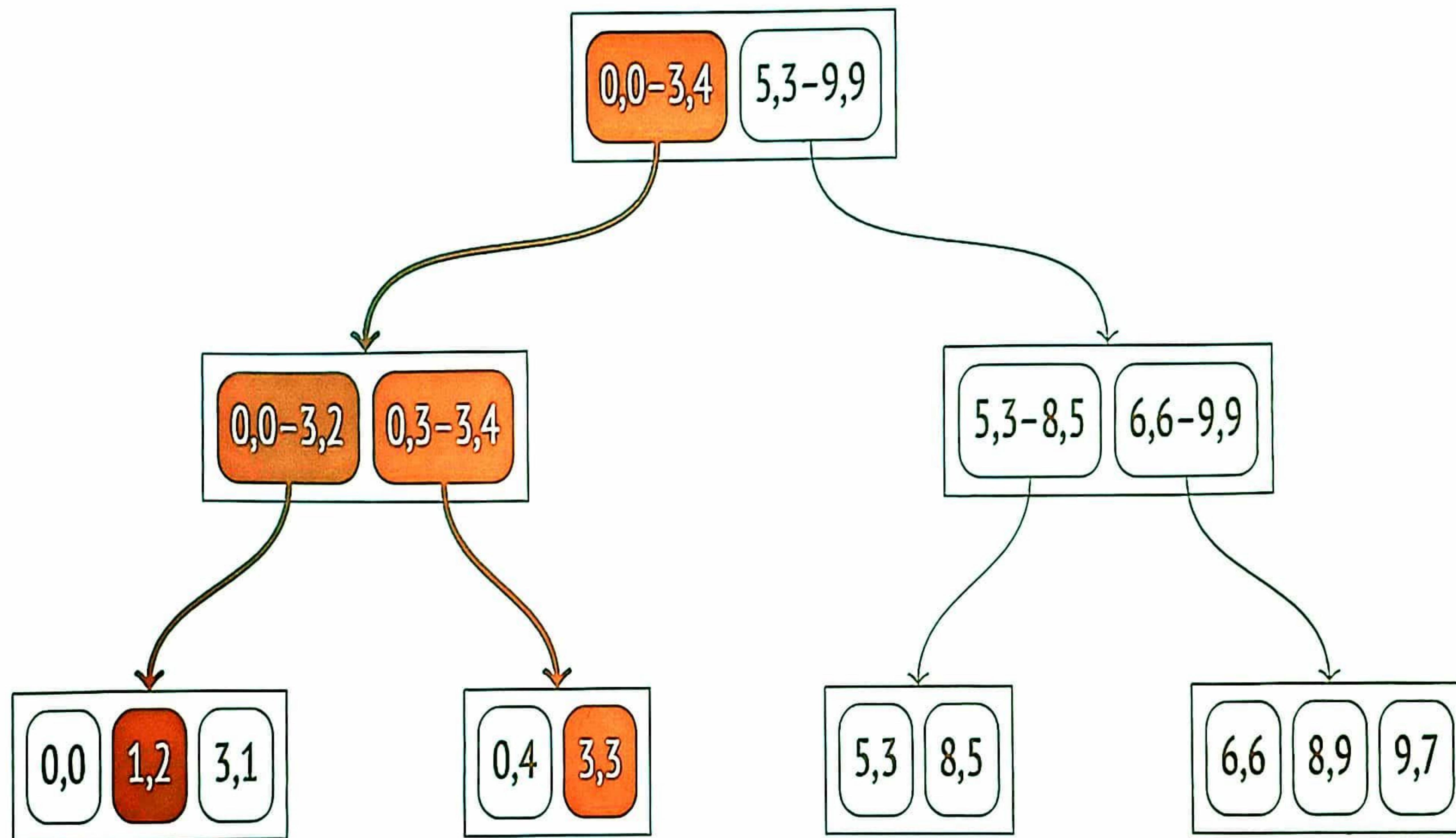
- Идея R-дерева состоит в том, что плоскость разбивается на прямоугольники, которые в сумме покрывают все индексируемые точки.
- Индексная запись хранит ограничивающий прямоугольник, а предикат можно сформулировать так: точка лежит внутри данного ограничивающего прямоугольника.

# Индекс k-NN



# Индекс

## R-дерево



# **Индекс**

## **RD-дерево для полнотекстового поиска**

- выбрать из набора документов те, которые соответствуют поисковому запросу.
- Для целей поиска документ приводится к специальному типу `tsvector`, который содержит лексемы и их позиции в документе.

# Индекс

## Полнотекстовый поиск

- Чтобы полнотекстовый поиск работал быстро, его нужно поддерживать индексом. Поскольку индексируются не сами документы, а значения типа `tsvector`, есть два варианта: построить индекс по выражению, с приведением типа, или создать отдельный столбец типа `tsvector` и индексировать его.

# Индекс

## RD-дерево

- R-дерево как таковое не годится для индексации документов, поскольку к ним неприменимо понятие ограничивающего прямоугольника. Используется модификация этого подхода – RD-дерево (Russian Doll, матрешка).
- Вместо ограничивающего прямоугольника это дерево использует ограничивающее множество, то есть множество, содержащее все элементы дочерних множеств.

# Индекс

## Сигнатурное дерево

- В ней используется фильтр Блума. Каждую лексему можно представить своей сигнатурой: битовой строкой определенной длины, в которой все биты равны нулю (сброшены), кроме одного, который равен единице (установлен). Номер установленного бита определяется значением хеш-функции от лексемы.

# **Индекс**

## **Аналогия**

- предметный указатель в конце книги. В указателе собраны все важные термины, и для каждого приведен список страниц, на которых этот термин упоминается. Чтобы указателем было удобно пользоваться, он составляется по алфавиту, иначе в нем невозможно было бы быстро ориентироваться. Так и GIN полагается на то, что элементы составных значений можно упорядочить, и в качестве основной структуры использует b-tree

# Индекс GIN

