

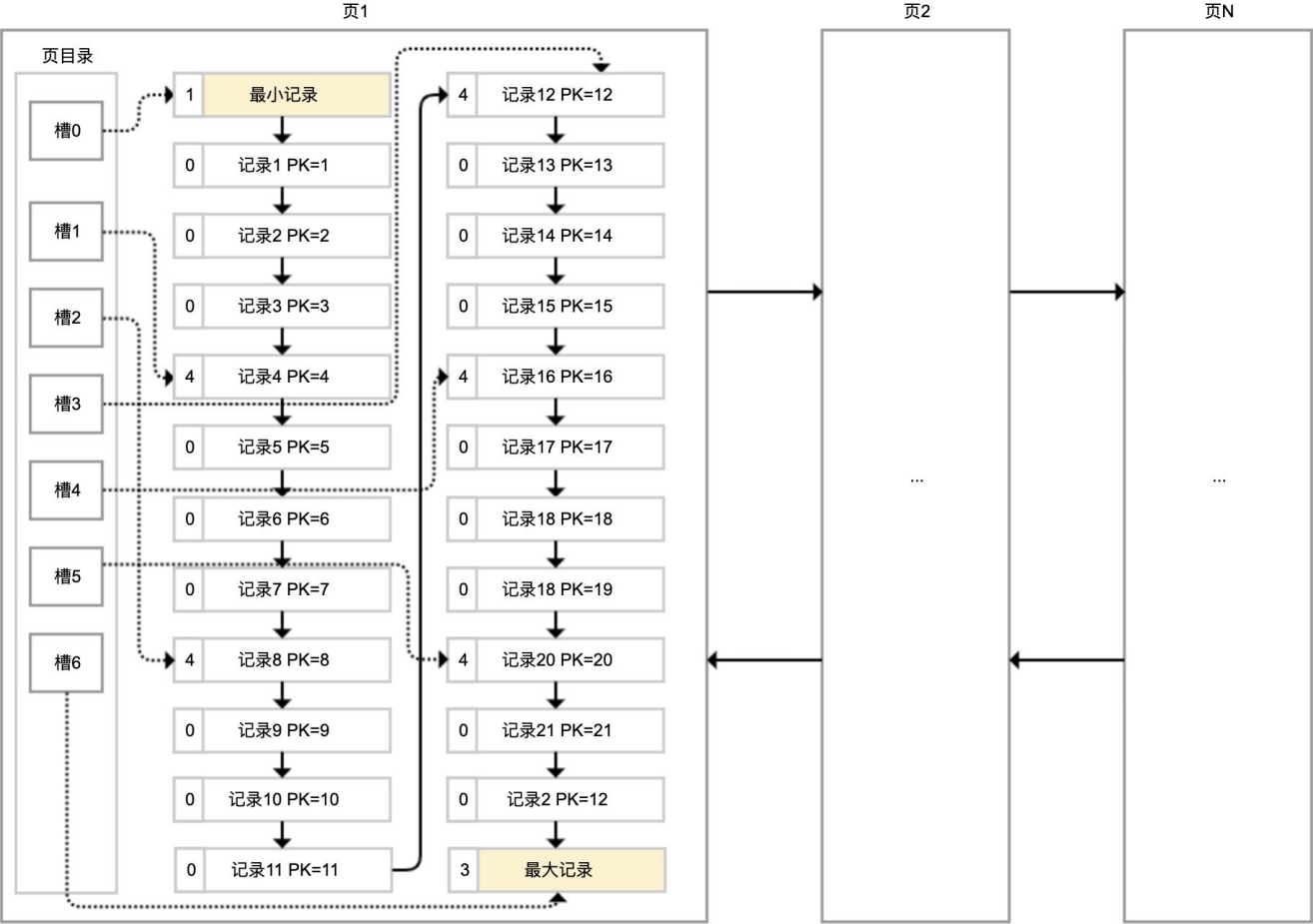
RDBMS Index

12/18/2022

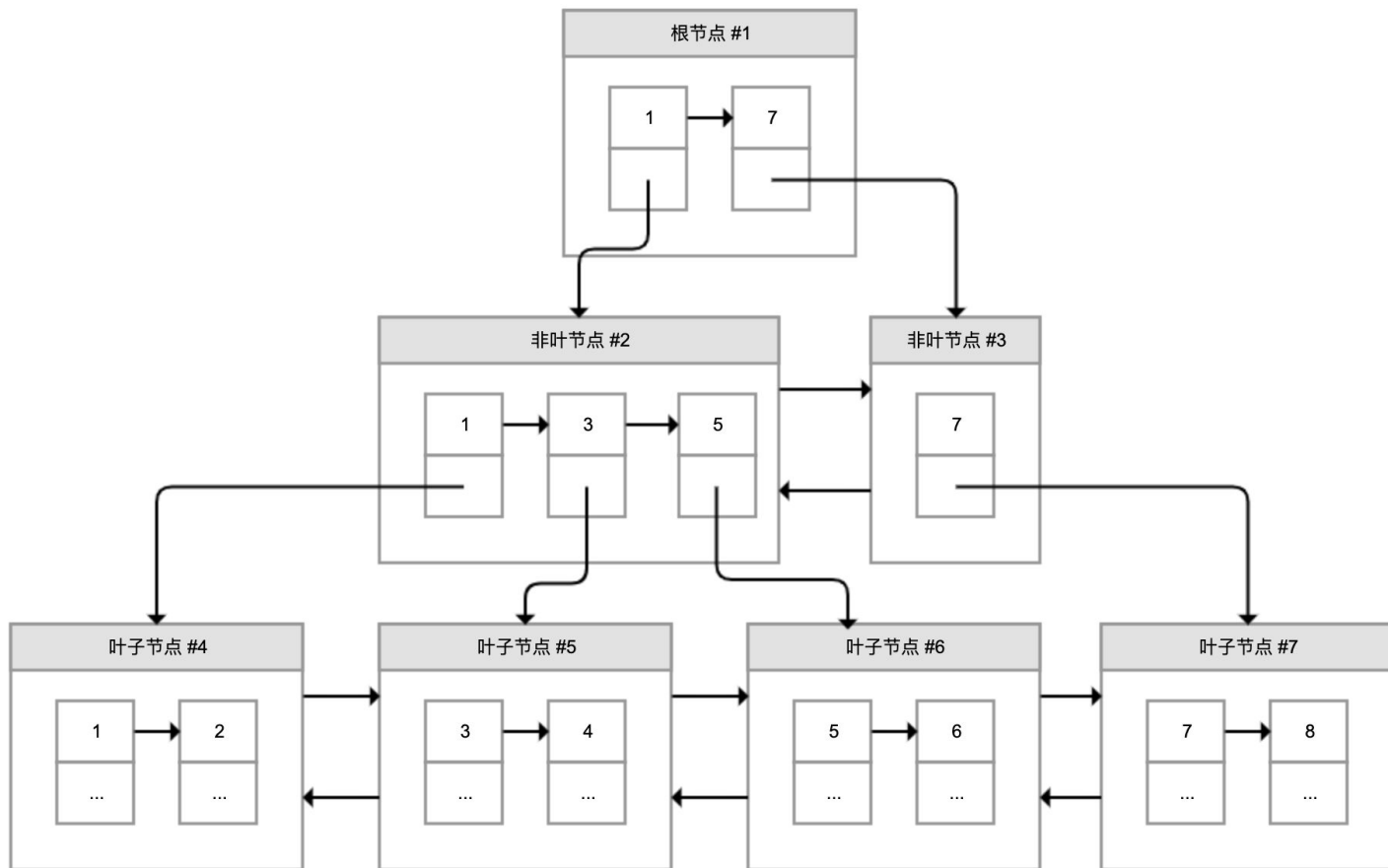
Index

- An additional structure derived from primary data and affects performance of queries.
- Speed up queries but slow down writes.

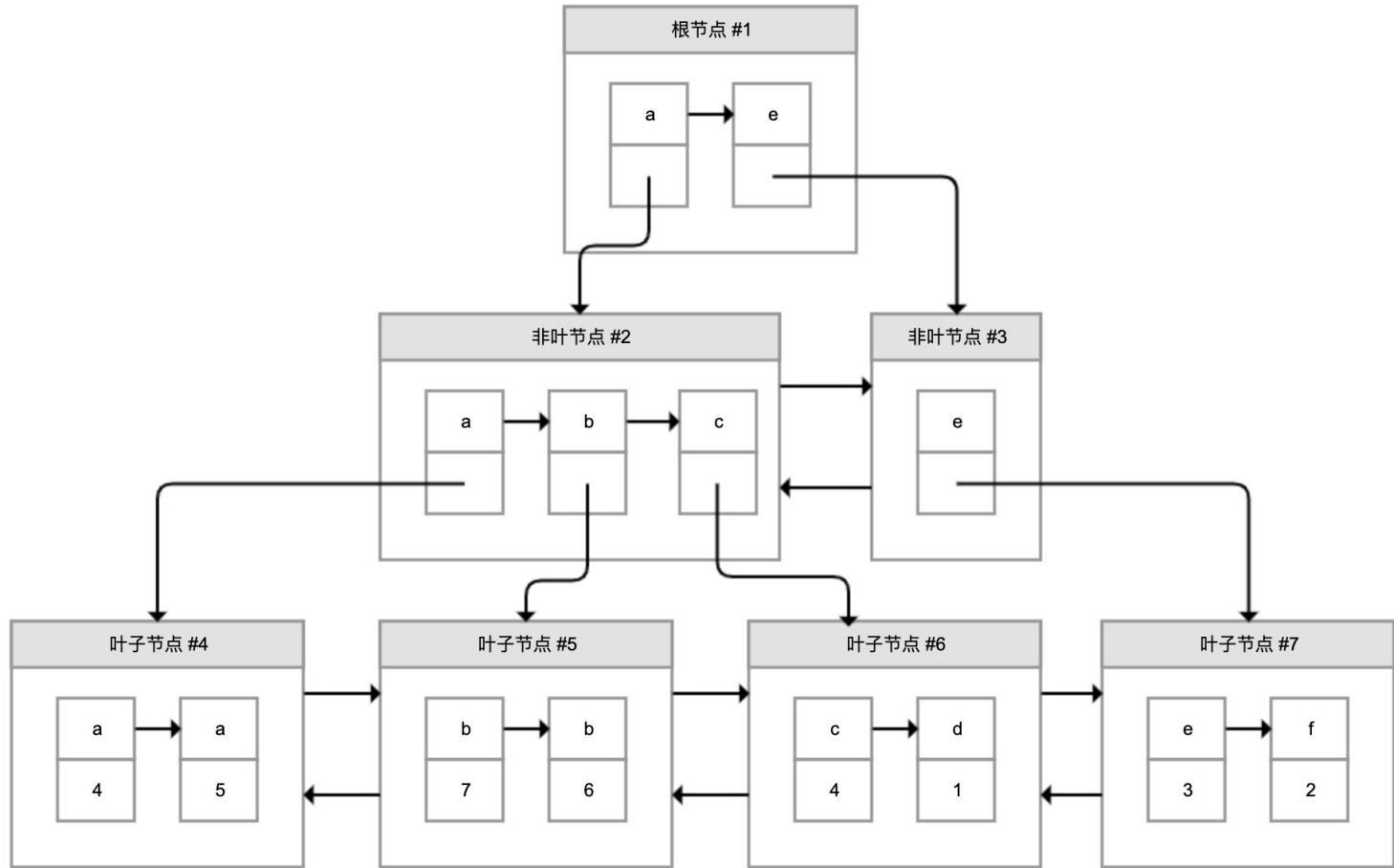
InnoDB如何存储数据



B+ Tree: Clustered Index



B+ Tree: Non - Clustered Index



Non - Clustered Index

- 维护代价 - 创建时的耗时
- 空间代价 - 创建后占用的空间
- 回表([retrieves rows from a table using an index](#))代价 - 查询时使用**辅助索引**先查询到数据行的主键id, 再**回到**聚簇索引中根据id查到真实的数据行;

Best Practice Summary 1

- 1) 无需一开始就建立索引, 可以等到业务场景明确后, 或者是数据量超过某个值(例如1万)、查询变慢后, 再针对需要查询、排序或分组的字段创建索引。创建索引后可以使用 EXPLAIN 命令, 确认查询是否可以使用索引。
- 2) 尽量索引轻量级的字段, 比如能索引 int 字段就不要索引 varchar 字段。索引字段也可以是部分前缀, 在创建的时候指定字段索引长度。针对长文本的搜索, 可以考虑使用 Elasticsearch 等专门用于文本搜索的索引数据库。
- 3) 尽量不要在 SQL 语句中 SELECT *, 而是 SELECT + 必要的字段, 甚至可以考虑使用 joint index 来包含要搜索的字段, 既能实现索引加速, 又可以避免回表的开销。

数据库如何决定是否用索引

- IO成本
- CPU成本

两个问题

- 是不是建了索引一定可以用上？
 - 用%进行前后缀匹配
 - Where后面的condition等值条件
 - Joint索引只用部分字段
- 怎么选择创建联合索引还是多个独立索引？
 - 根据常用查询使用的字段

References

Optimisation: <https://dev.mysql.com/doc/refman/5.6/en/index-condition-pushdown-optimization.html>