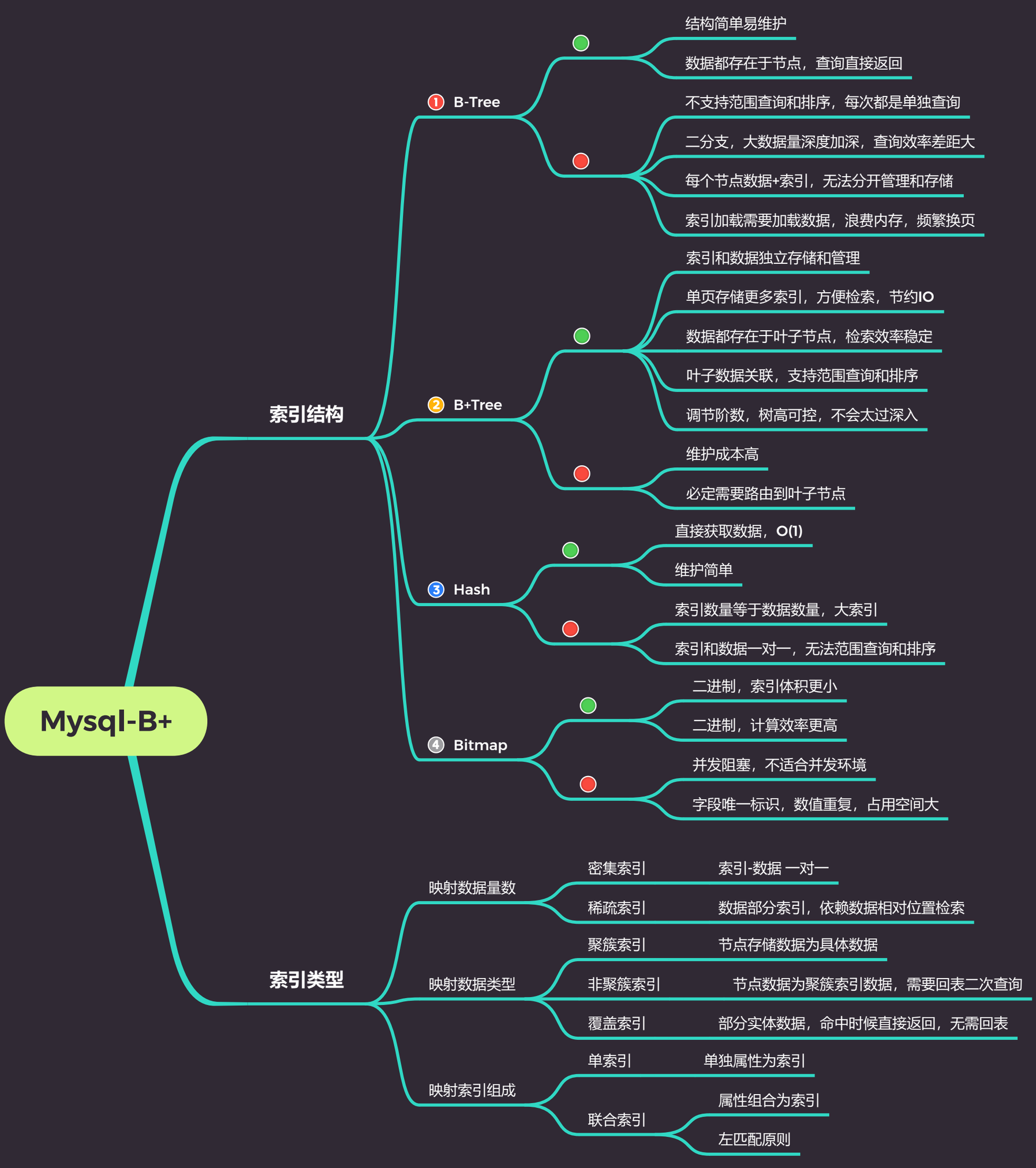
Mysql-B+
索引结构
1 B-Tree
结构简单易维护
数据都存在于节点，查询直接返回
不支持范围查询和排序，每次都是单独查询
二分支，大数据量深度加深，查询效率差距大
每个节点数据+索引，无法分开管理和存储
索引加载需要加载数据，浪费内存，频繁换页
2 B+Tree
索引和数据独立存储和管理
单页存储更多索引，方便检索，节约IO
数据都存在于叶子节点，检索效率稳定
叶子数据关联，支持范围查询和排序
调节阶数，树高可控，不会太过深入
维护成本高
必定需要路由到叶子节点
3 Hash
直接获取数据，O(1)
维护简单
索引数量等于数据数量，大索引
索引和数据一对一，无法范围查询和排序
4 Bitmap
二进制，索引体积更小
二进制，计算效率更高
并发阻塞，不适合并发环境
字段唯一标识，数值重复，占用空间大
索引类型
映射数据量数
密集索引
索引-数据 一对一
稀疏索引
数据部分索引，依赖数据相对位置检索
映射数据类型
聚簇索引
节点存储数据为具体数据
非聚簇索引
节点数据为聚簇索引数据，需要回表二次查询
覆盖索引
部分实体数据，命中时候直接返回，无需回表
映射索引组成
单索引
单独属性为索引
联合索引
属性组合为索引
左匹配原则