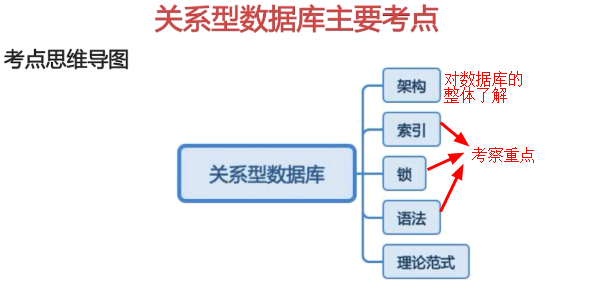
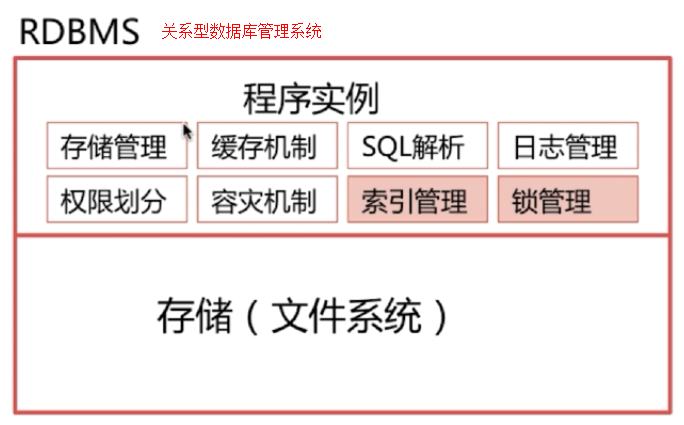
**03 数据库**





程序实例模块：对存储进行逻辑上的管理

存储管理模块：数据的逻辑关系转换成物理存储关系

缓存机制模块：优化执行效率

SQL解析模块：解析SQL语句

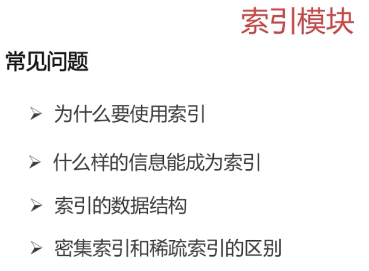
日志管理模块：记录操作

权限管理模块：进行多用户操作权限管理

容灾机制模块：进行灾难恢复

索引模块：优化数据查询效率

锁模块：使得数据库支持并发操作



为什么要使用索引？

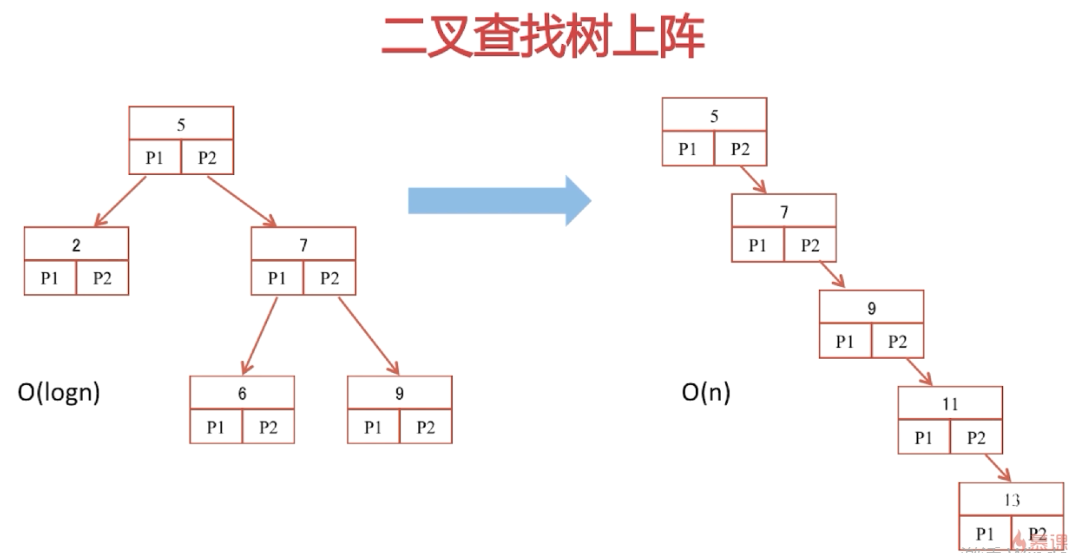
答：为了避免在大规模表中使用全表搜索。基于类似字典的原理，依据关键信息，根据一定查找方式（索引），可以大幅提高查询速度。

什么信息能成为索引？

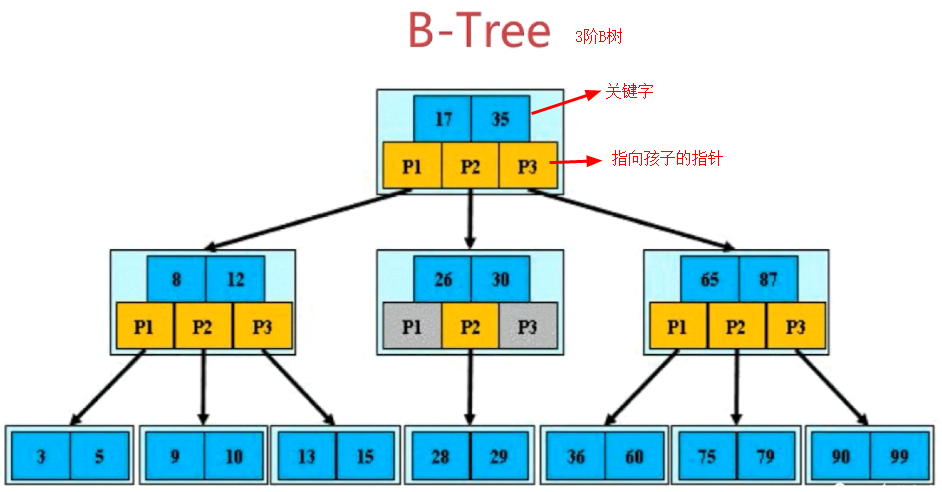
答：主键、唯一键及普通键（？）等

索引的数据结构？

答：（1）二叉查找树、平衡二叉查找树（2）B树（3）B+树（4）Hash结构



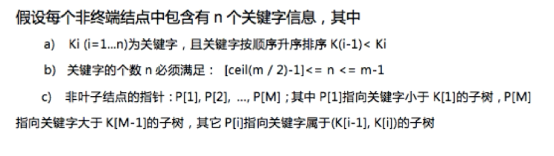
二叉查找树可以减少查找的时间复杂度，但可能会退化成线性深度和线性时间复杂度。同时较多的IO操作也会降低时间效率。



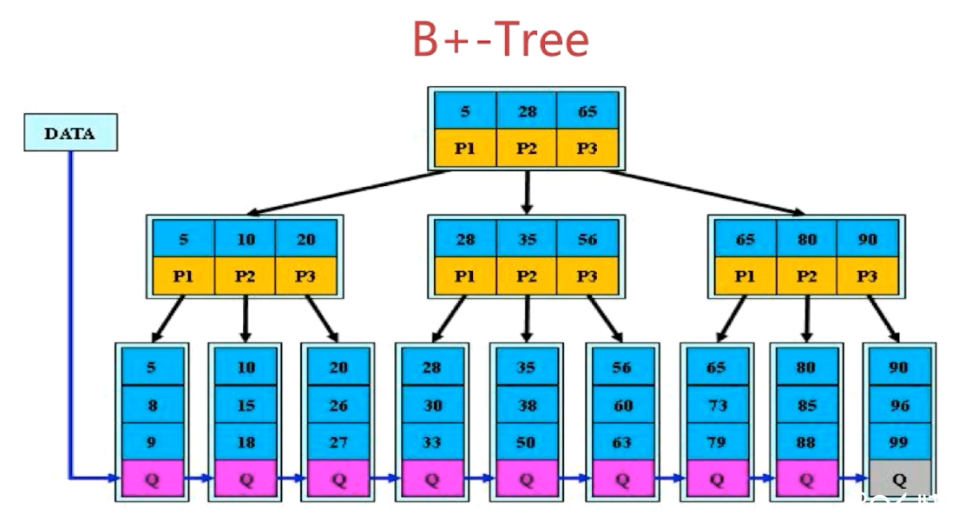
B树（平衡多路查找树），每个节点最多有M个孩子的树，叫做M阶B树。B树遵循的约束：

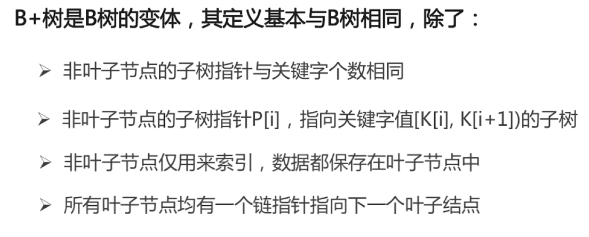
1. 根节点至少包括两个孩子
2. 树中每个节点最多含有m个孩子（m>=2）
3. 除根节点和叶子节点外，其他节点至少有ceil(m/2)个孩子
4. 所有叶子节点都位于同一层

目的是让每个索引块存储更多的信息，降低树的高度，减少IO次数

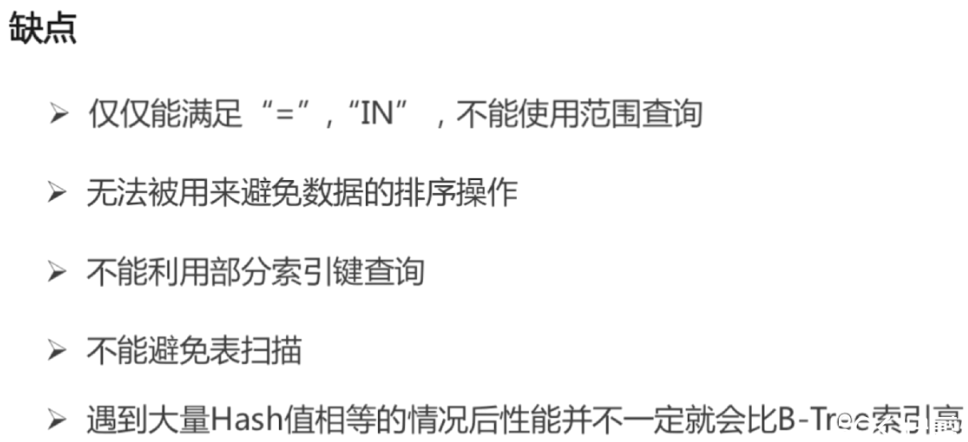


B+树：



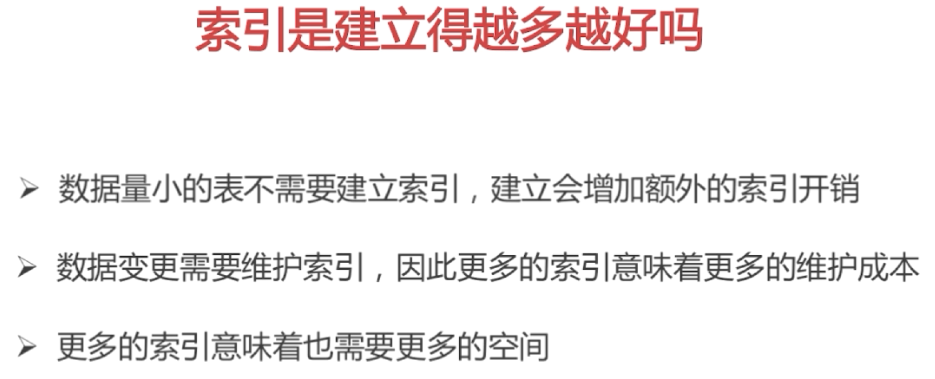


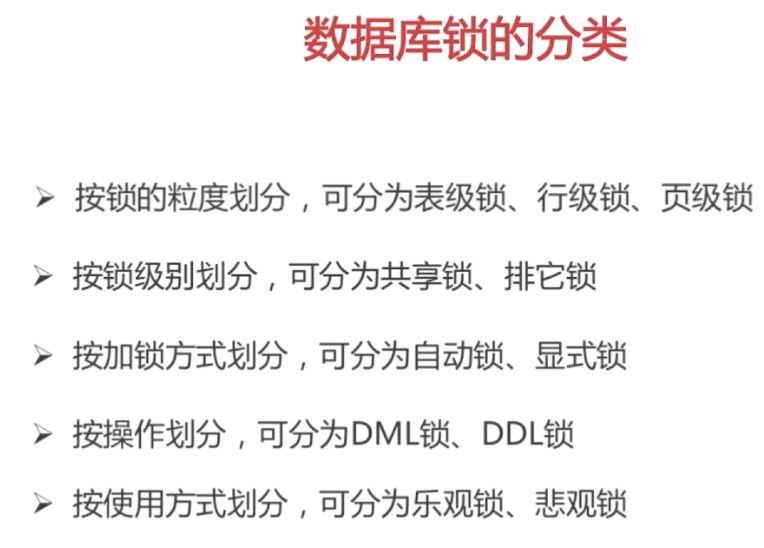
Hash 索引——只需经过hash函数运算，一次定位，避免多次IO操作：



如何优化慢查询SQL？

答：具体场景的经验题，没有经验可以说不知道





**数据库事务的四大特性（ACID）**：

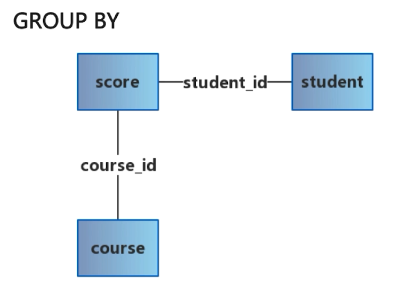
1. 原子性（atomicity）——数据库系统保证一个事务中的所有操作要么全部成功，要么全部失败
2. 一致性（consistency）——事务必须确保数据库在事务开始时和事物结束时都处于一个正确一致的状态。例如不能破坏参照一致性约束、两转账账户总款数不变等
3. 隔离性（isolation）——一个事务提交之前，事务中对数据库的修改对其他查询和事务都是隔离的
4. 持久性（durability）——事务提交后，在事务中做的所有修改都是永久的。在数据库崩溃后可恢复的的。使得数据库的当前状态永不丢失

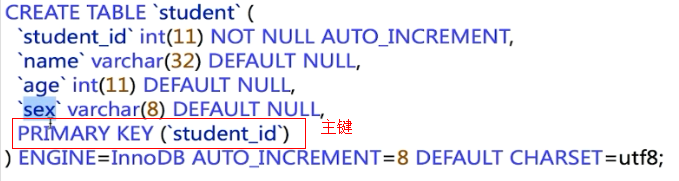
**SQL语法：**

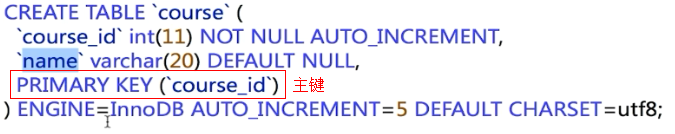
一些基础语法：INSERT、SELECT、WHERE、INNER JOIN、LEFT OUT JOIN 等

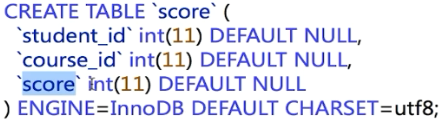
一些关键语法：GROUP BY、HAVING、统计相关：COUNT（统计）、SUM(求总和)、MAX、MIN、AVG

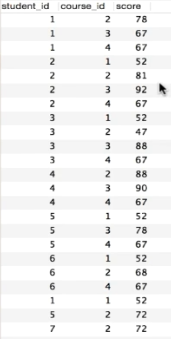
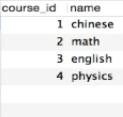
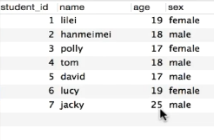
举例用的表结构：







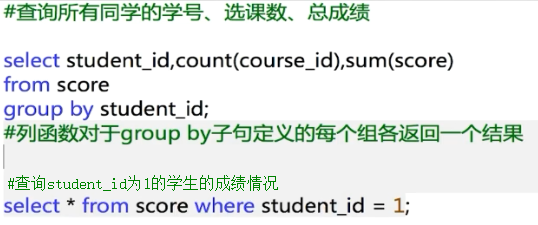


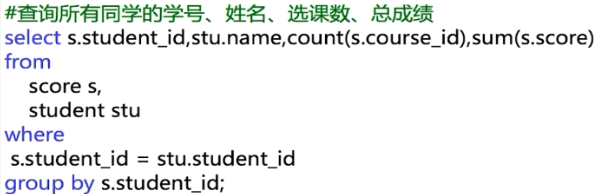
  

Score table course table student table

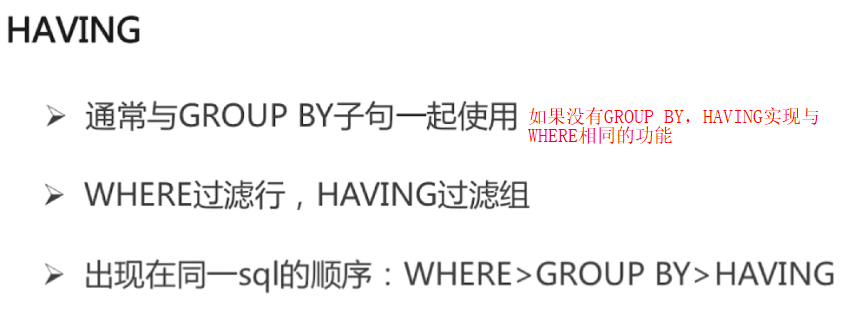
**GROUP BY**:根据给定数据列的每个成员，对查询结果分组统计，得到一个分组汇总表。SELECT子句的列名必须为分组或列函数（该限定只针对同一张表成立）

SQL代码示例：





**HAVING:**



SQL代码示例：



