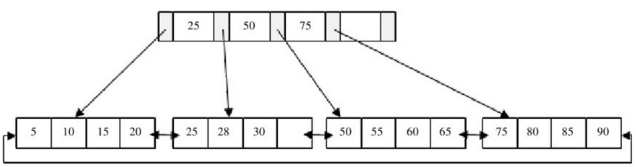
## 存储引擎

Innodb

myisam

## 索引：

1. **B+树**



非叶子节点不存放data数据，只起索引的作用，叶子节点会存放最终的数据，而且叶子节点之间通过双向链表进行链接，可以由当前节点访问前后的节点。

**2. InnoDB中的聚集索引**

InnoDB存储引擎表是索引组织表，即表中数据按照主键顺序存放，而聚集索引就是按照每张表的主键构建一颗B+树，同时叶子节点中存放的表中一行的数据，也将聚集索引的叶子节点称为数据页。通过聚集索引可以直接在叶子节点上获取数据，此外，聚集索引可以特别快的进行范围查询。

**3. 辅助索引**

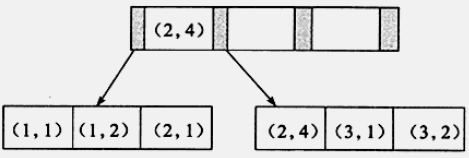
Innodb中除了聚集索引外其他索引称为辅助索引，辅助索引也是B+树的结构，与聚集索引不同的是辅助索引叶子节点存放的不是具体的某一行的数据，而是相关联的那一行的主键，也就是说，如果我们通过辅助索引查询之后得到主键，还需要去聚集索引那边根据主键来查询具体的行。

**4. Cardinality**

Cardinality中文名为基数，又称为索引选择性，索引选择性计算公式为某列中的非重复值的个数/该列所有值的个数，结果的取值范围为[0-1]，越高代表选择性越高，也就代表着索引的效果越好。例如性别属性，在一般表中男女比例为1:1，如果给性别属性加上索引，其索引选择性为0.5，相当于只能区分百分之五十的数据，这个时候才用索引完全没有必要。

**5. 联合索引**

联合索引是对多个列进行匹配，例如有列a，b，建立联合索引（a，b），其B+树为：



可以看到，索引的顺序首先按照a的值进行排序，然后在同一个a的情况下，按照b的顺序进行排序。

对于查询SELECT \* FROM TABLE WHERE a=xxx and b=xxx，是可以使用（a,b）这个联合索引的，对于单个a列查询SELECT \* FROM TABLE WHERE a=xxx，也是可以使用联合索引，因为索引的顺序就是a的顺序，而对于单个b列是不可以使用联合索引，因为联合索引中b并没有顺序。这就是联合索引的最左匹配原则。

联合索引的第二个好处就是已经对第二个键进行了排序，当第一个键确定下来后，order by第二个键的话可以直接获取，不需要进行排序操作。

## 优化