

【实验】索引

【实验目的】

【实验准备】

- 索引定义;
- 查询索引;

```
show index from 表名;  
  
select distinct table_name,index_name from information_schema.statistics  
where table_schema='数据库名称';
```

- 索引的优势及劣势;
- BTREE索引;
- HASH索引;
- R-Tree索引 (空间索引) ;
- Full-Text (全文索引) ;

索引	INNODB引擎	MYISAM引擎	MEMORY引擎
BTREE索引	支持	支持	支持
HASH 索引	不支持	不支持	支持
R-tree 索引	不支持	支持	不支持
Full-text	5.6版本之后支持	支持	不支持

索引定义：索引（index）是帮助MySQL高效获取数据的数据结构（有序）。索引是在数据库表的字段上添加的，是为了提高查询效率存在的一种机制。在数据之外，数据库系统还维护着满足特定查找算法的数据结构，这些数据结构以某种方式引用（指向）数据，这样就可以在这些数据结构上实现高级查找算法，这种数据结构就是索引。

简而言之，索引就是一个排好序的数据结构。

索引的优势：

- 加快查找和排序的速率，降低数据IO成本及CPU消耗；
- 通过创建唯一性索引，保证数据库表每一行数据的唯一性。

索引的优势及劣势：

- 索引也是一张表，该表保存了主键和索引字段，且指向实体类记录，它也需要占用空间的；
- 通过索引可以增加查询效率，但对修改数据(增|删|改)操作，每次修改都需要更新一次索引
- BTREE索引：MySQL中，采用B树这种数据结构进行存储，这也是最常见的索引类型
- HASH索引：只有Memory引擎支持，使用场景简单。

- **R-Tree**索引（空间索引）：空间索引是MyISAM引擎的一个特殊索引类型，主要用于地理空间数据类型，通常应用场景较少。
- **Full-Text**（全文索引）：全文索引也是MyISAM的一个特殊索引类型，主要用于全文索引，InnoDB从Mysql5.6版本开始支持全文索引。

平常所说的索引，如果没有特别指明，都是指B+树（多路搜索树，并不一定是二叉的）结构组织的索引。其中聚集索引、复合索引、前缀索引、唯一索引默认都是使用 B+tree 索引，统称为 索引。

【实验内容】

- 为employees表的age字段添加索引；
- 查询job_title为"程序员"的员工信息；
- 为employees表的department_id字段添加外键约束；
- 查询salary大于10000的员工数量；
- 查询department_id为1或2的员工信息；
- 查询所有员工的平均年龄；
- 查询department_name为"市场部"的部门编号和员工数量；
- 为employees表的gender字段添加索引；
- 查询salary排名前10的员工信息；
- 查询age大于35且job_title为"程序员"的员工信息；
- 查询每个部门的平均工资和最高工资；
- 为employees表的salary字段添加索引；
- 查询manager_id为101的员工信息；
- 查询每个部门的员工数量和平均年龄；
- 为employees表的department_id、job_title和salary字段添加联合索引；