# 数据页

## 概述

要了解页，需要了解局部性原理：从磁盘读取数据的时候，不是按照你需要的实际大小取数据，而是按照页为单位取数据（避免频繁的磁盘访问）。

页是InnoDB管理存储空间的基本单位，一个页的大小默认是16KB（操作系统默认是4KB，不要混淆）：

SHOW GLOBAL STATUS like ‘Innodb\_page\_size’;

## 结构



### File Header

### Page Header

### Infimum+Supremum Records

### User Records

### Free Space

### Page Directory

### File Trailer

## 数据插入/查找



基本过程：

1. 首先第一条数据插入到用户数据区域；
2. 第二条数据插入前，首先进行排序（按照主键），然后确定插入位置（不一定第二条数据就在第一条数据后面）；
3. 以此类推；

注：这样的插入操作并不高，因为需要额外的排序，这样做的目的是提高查询的速度。

1. 查询的时候，先把数据所在的那一页数据全部从磁盘读取到内存，然后从记录中依次查找（但是极端情况下可能会遍历所有行数据的链表）；
2. 此时需要页目录进行快速定位（目录会用专门的算法对用户行数据进行分组），目录项中存储的是该组中主键（或索引）的最小值；
3. 这样查找数据的时候，先根据值的大小从目录项查找，然后具体定位到组，进而查找用户行数据；如果在对应组中没有找到，则不需要继续查找；
4. 如果数据非常大，则需要多个页：



1. 如果存储数据非常大，则页也非常多，页的链表很长，极端情况下数据查找可能会遍历整个页的链表，此时需要一个管理页的结构（每个结构存储所管理的页的最小值）：



上面叫目录页，主要存储的是目录的信息，下面的数据页，主要存储的是具体的数据。

数据页的next指针用于大于>操作，prev指针用于小于<操作（这与真正的B+树有所不同，严格的B+树只有next指针）。

# 索引