

## 第二十三讲：InnoDB 行记录格式的演进过程

知春路遇上八里桥

<2024-08-14 Wed>



## 1 背景知识

## 2 底层细节



1

## 背景知识



# 行记录格式<sup>1</sup>

## ① REDUNDANT (冗余)

- ▶ 在 MySQL 5.0.3 之前的版本中, InnoDB 默认使用 REDUNDANT 格式
- ▶ 它在每个记录中都保存了完整的字段信息, 包括隐藏列, 这使得它相对更占用空间

## ② COMPACT (紧凑)

- ▶ COMPACT 是 MySQL 5.0.3 版本之后 InnoDB 的默认行记录格式
- ▶ 通过更紧凑的方式来存储数据, 减少了记录间的额外空间, 提高了空间利用率
- ▶ 对于长字段, COMPACT 格式会将前 768 字节存储在数据页中, 剩余的数据则存储在溢出页中
- ▶ 对于 NULL 值, COMPACT 格式不会占用额外的空间
- ▶ 并且对于一些短字段类型, 它会使用更少的字节来存储

## ③ DYNAMIC (动态)

- ▶ DYNAMIC 格式在 MySQL 5.7.9 版本引入, 是 COMPACT 的变体, 提高了对长文本的处理能力
- ▶ 对于长字段, DYNAMIC 格式会将数据完全放到溢出页, 索引内只留一个 20 字节的指针
- ▶ 这种方式显著减少了数据页的空间占用, 特别是在处理大量长文本或 BLOB 字段时

## ④ COMPRESSED (压缩)

- ▶ COMPRESSED 格式是在 MySQL 5.6.4 版本中引入的, 它允许 InnoDB 表在行级别进行压缩
- ▶ 减少磁盘 I/O 和存储空间的需求, 解压缩会增加 CPU 的负担
- ▶ 并非所有 InnoDB 表都可以使用 COMPRESSED 格式, 可能受到一些限制, 如索引键长度等

<sup>1</sup><https://dev.mysql.com/doc/refman/8.0/en/innodb-row-format.html>



# 行格式相关语句

- 建表时指定行格式，或者直接修改表的行格式

```
CREATE TABLE t1 ( /* 表结构定义 */ ) ROW_FORMAT=REDUNDANT;  
ALTER TABLE t1 ROW_FORMAT=REDUNDANT/DYNAMIC;
```

- 查看表的状态中可以知道行格式，例如：

```
mysql> show table status like '%t1%'\G  
***** 1. row *****  
      Name: t1  
      Engine: InnoDB  
      Version: 10  
      Row_format: Dynamic  
      Rows: 2  
      Avg_row_length: 8192  
      Data_length: 16384  
      Max_data_length: 0  
      Index_length: 0  
      Data_free: 0  
      Auto_increment: NULL  
      Create_time: 2024-08-12 20:48:23  
      Update_time: 2024-08-12 21:40:47  
      Check_time: NULL  
      Collation: utf8mb4_0900_ai_ci  
      Checksum: NULL  
      Create_options:  
      Comment:  
1 row in set (0.01 sec)
```

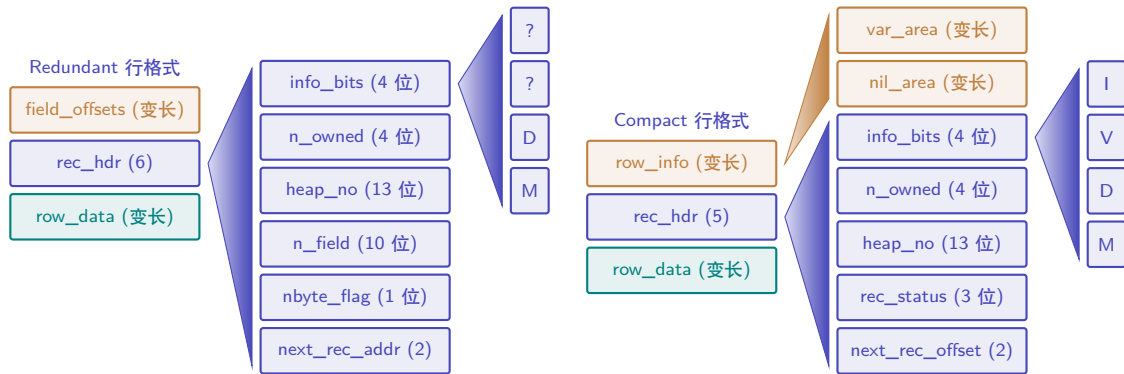


2

## 底层细节



# REDUNDANT vs COMPACT



- n\_field 表示记录的数量
- nbyte\_flag 标记 field\_offsets 列表中每个长度占 1 个字节
  - ▶ nbyte\_flag=1 时, 占 1 个字节
  - ▶ nbyte\_flag=0 时, 占 2 个字节
- field\_offsets 长度的最高位如果为 1 表示当前数据位 NULL



# 实验演示





# 结束

