

第二十四讲：MySQL 的 Online DDL 的演进过程

知春路遇上八里桥

<2024-08-19 Mon>



1 背景知识

2 特性介绍



1

背景知识



Online DDL

- 日常需求: DDL 语句修改表结构, 添加列, 删除列, 修改列的类型等
- 表的 ONLINE DDL¹ 的 ALGORITHM=COPY/INPLACE
- ALGORITHM=COPY 算法
 - ① 按照原表 t1 定义创建一个新的临时表 t2
 - ② 对原表 t1 加写锁 (禁止 DML, 允许 SELECT)
 - ③ 建立的临时表 t2 中执行 DDL
 - ④ 将原表 t1 中的数据 COPY 到临时表 t2
 - ⑤ 释放原表 t1 的写锁
 - ⑥ 将原表 t1 删除, 并将临时表 t2 重命名为原表 t1
- ALGORITHM=INPLACE 算法, 减少了磁盘空间的占用和数据的复制成本
 - ▶ 直接在原表 t1 上执行 DDL 操作, 无需创建临时表
 - ▶ 在原表 t1 上进行更改, 不需要生成临时表。根据是否变更行记录格式, 分为两类:
 - ① rebuild: 需要重建表 (重新组织聚簇索引)。比如添加索引、添加/删除列等
 - ② no-rebuild: 不需要重建表。只需要修改表的元数据, 比如删除索引、修改列名等
 - ▶ 对于 rebuild 方式实现 ONLINE 是通过缓存 DDL 期间的 DML
 - ① 待 DDL 完成之后, 将 DML 应用到原表 t1 上来实现的

¹<https://dev.mysql.com/doc/refman/8.0/en/innodb-online-ddl-operations.html>



Instant DDL 改进

- MySQL 8.0.12 实现了 ALTER TABLE ... ADD COLUMN 采用 ALGORITHM=INSTANT 选项
 - ▶ 仅支持添加列，并且有一些限制
 - ① 列只能添加到表的最后，不能插在其他列之间
 - ② 如果不重建表的话，不运行添加主键列 (PK)
 - ③ 添加的列不支持压缩格式
 - ▶ 2018 WL#11250 : Support Instant Add Column | worklog | blog
- MySQL 8.0.29 改进了 ALGORITHM=INSTANT 的实现，使其变得更加灵活
 - ▶ 支持删除列

```
ALTER TABLE t1 DROP COLUMN c3, ALGORITHM=INSTANT;
```
 - ▶ 支持添加或删除到任何位置的列

```
ALTER TABLE t1 ADD COLUMN c3 VARCHAR(10) AFTER c1, ALGORITHM=INSTANT;
```

```
ALTER TABLE t1 ADD COLUMN c3 VARCHAR(10) FIRST, ALGORITHM=INSTANT;
```
 - ▶ 2023 INSTANT ADD and DROP Column | post1, post2
- ALGORITHM=INSTANT 可以使得 DDL 和表重建 (table rebuild) 分离
 - ▶ ALTER TABLE 时不 COPY 表中的数据
 - ▶ OPTIMIZE TABLE 才重建表



2

特性介绍



变更观测

查询过程如下, name 和 total_row_versions 记录了列的版本信息

```
mysql> create table t1 (c1 char(10));  
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> SELECT NAME,TOTAL_ROW_VERSIONS FROM INFORMATION_SCHEMA.INNODB_TABLES WHERE NAME LIKE "%t1%";  
+-----+-----+  
| NAME      | TOTAL_ROW_VERSIONS |  
+-----+-----+  
| test/t1   | 0                  |  
+-----+-----+  
1 row in set (0.01 sec)
```

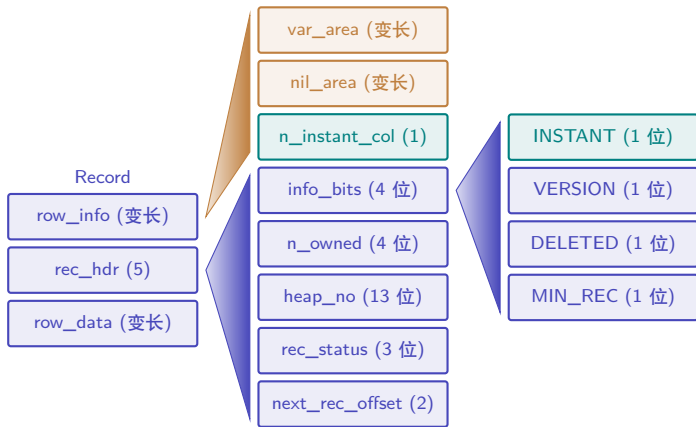
```
mysql> alter table t1 add column c0 char(10) first, algorithm=instant;  
Query OK, 0 rows affected (0.05 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> SELECT NAME,TOTAL_ROW_VERSIONS FROM INFORMATION_SCHEMA.INNODB_TABLES WHERE NAME LIKE "%t1%";  
+-----+-----+  
| NAME      | TOTAL_ROW_VERSIONS |  
+-----+-----+  
| test/t1   | 1                  |  
+-----+-----+  
1 row in set (0.01 sec)
```



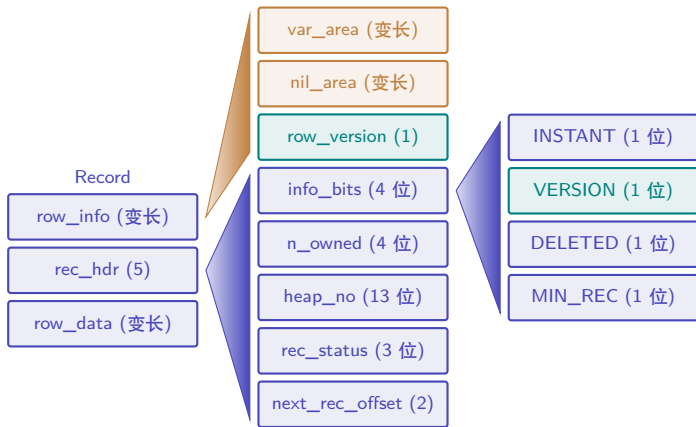
INSTANT 标记方案

- 开启 `info_bits` 的 INSTANT 标志位
- `row_info` 中插入一个字节记录行的包含的列数 `n_instant_col`



VERSION 标记方案

- 开启 info_bits 的 VERSION 标志位
- row_info 中插入一个字节记录行的版本信息 row_version



结束

