MySQL执行计划相关笔记

使用EXPLAIN关键字可以模拟优化器执行SQL语句，分析你的查询语句或是结构的性能瓶颈，在select语句之前增加 explain 关键字，MySQL 会在查询上设置一个标记，执行查询会返回执行计划的信息，而不是执行这条SQL

注意：如果from中包含子查询，仍会执行该子查询，将结果放入临时表中

示例表：

DROP TABLE IF EXISTS `actor`;

CREATE TABLE `actor` (

`id` INT (11) NOT NULL,

`name` VARCHAR (45) DEFAULT NULL,

`update\_time` datetime DEFAULT NULL,

PRIMARY KEY (`id`)) ENGINE = INNODB DEFAULT CHARSET = utf8;

INSERT INTO `actor` (`id`,`name`,`update\_time`) VALUES (1,'a','2017-12-22 15:27:18'),(2,'b','2017-12-22 15:27:18'),(3,'c','2017-12-22 15:27:18');

DROP TABLE IF EXISTS `film`;

CREATE TABLE `film` (

`id` INT (11) NOT NULL AUTO\_INCREMENT,

`name` VARCHAR (10) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `idx\_name` (`name`)

) ENGINE = INNODB DEFAULT CHARSET = utf8;

INSERT INTO `film` (`id`,`name`) VALUES (3,'film0'),(1,'film1'),(2,'film2');

DROP TABLE IF EXISTS `film\_actor`;

CREATE TABLE `film\_actor` (

`id` INT (11) NOT NULL,

`film\_id` INT (11) NOT NULL,

`actor\_id` INT (11) NOT NULL,

`remark` VARCHAR (255) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `idx\_film\_actor\_id` (`film\_id`,`actor\_id`)

) ENGINE = INNODB DEFAULT CHARSET = utf8;

INSERT INTO `film\_actor` (`id`,`film\_id`,`actor\_id`) VALUES (1,1,1),(2,1,2),(3,2,1);

执行语句：

explain select \* from actor;

在查询中的每个表会输出一行，如果有两个表通过join连接查询，那么会输出两行

explain两个变种

explain extended：会在explain的基础上额外提供一些查询优化的信息。紧随其后通过 show warnings 命令可以得到优化后的查询语句，从而看出优化器优化了什么。额外还有 filtered列，是一个半分比的值，rows\*filtered/100可以估算出将要和explain中前一个表进行连接的行数（前一个表指explain中的id值比当前表id值小的表）。

语句：

explain extended select \* from film where id = 1;

explain partitions：相比explain多了个partitions字段，如果查询是基于分区表的话，会显示查询将访问的分区。

语句：

explain partitions select \* from film where id = 1;

show warnings; 执行计划后面跟着show warnings会得到数据库优化后的语句（不一定能执行）

explain中的列

1. id列

id列的编号是select的序列号，有几个select就有几个id，并且id的顺序是按select出现的顺序增长的。

id列越大执行优先级越高，id相同则从上往下执行，id为NULL最后执行。

1. select\_type列

select\_type 表示对应行是简单还是复杂的查询。

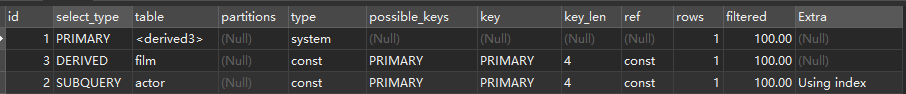
1. simple 简单查询。查询不包含子查询和union 例：explain select \* from film where id = 2;
2. primary 复杂查询中最外层的select
3. subquery 包含在select中的子查询（不在from子句中）
4. derived 包含在from子句中的子查询。MySQL会将结果存放在一个临时表中，也称为派生表（derived的英文含义）
5. union 在union中的第二个和随后的select

用这个例子来了解primary、subquery和derived类型

set session optimizer\_switch='derived\_merge=off'; #关闭mysql5.7新特性对衍生表的合并优化

explain select (select 1 from actor where id = 1) from (select \* from film where id = 1) der;

结果：



set session optimizer\_switch='derived\_merge=on'; #还原默认配置

1. table列

这一列表示explain的一行正在访问哪个表。

当from子句中有子查询时，table列是<derivenN>格式，表示当前查询依赖 id=N 的查询，于是先执行 id=N 的查询。

当有 union 时，UNION RESULT 的 table 列的值为<union1,2>，1和2表示参与 union 的 select 行id。

1. type列