

数据库的发展

在二十世纪六十年代，随着供应商开始销售，计算机化的物流技术，用于制造和更广泛的实验室使用。数据库管理系统（DBMS）的出现。DBMS或现代数据库允许用户编组大量的数据。在存储介质上组织记录，以实现最佳访问的艰巨任务。现在称为数据库管理系统的子系统处理。

当前数据库分为关系型数据库和非关系型数据库

关系型数据库

关系型数据库：指采用了关系模型来组织数据的数据库。关系模型指的就是二维表格模型，而一个关系型数据库就是由二维表及其之间的联系所组成的一个数据组织。关系型数据库的优点：1.容易理解：二维表结构是非常贴近逻辑世界的一个概念，关系模型相对网状、层次等其他模型来说更容易理解 2.使用方便：通用的SQL语言使得操作关系型数据库非常方便 3.易于维护：丰富的完整性(实体完整性、参照完整性和用户定义的完整性)大大减低了数据冗余和数据不一致的概率 主流数据库 Oracle, Microsoft SQL Server, MySQL, PostgreSQL, DB2, Microsoft Access, SQLite, Teradata, MariaDB(MySQL的一个分支), SAP

非关系型数据库

非关系型数据库：指非关系型的，分布式的，且一般不保证遵循ACID原则的数据存储系统。（ACID是Atomic（原子性） Consistency（一致性） Isolation（隔离性） Durability（持久性）） 非关系型数据库：Mongodb, redis是一个开源的、使用C语言编写的、支持网络交互的、可基于内存也可持久化的Key-Value数据库，hbase, leveldb

mysql数据类型

整型

tinyint(m) 1个字节 范围(-128~127) smallint(m) 2个字节 范围(-32768~32767)
mediumint(m) 3个字节 范围(-8388608~8388607) int(m) 4个字节 范围
(-2147483648~2147483647) bigint(m) 8个字节 范围(+/-9.22*10的18次方)

浮点型(float和double)

float(m,d) 单精度浮点型 8位精度(4字节) m总个数, d小数位 double(m,d) 双精度浮点型 16位精度(8字节) m总个数, d小数位

字符串(char,varchar,_text)

char(n) 固定长度, 最多255个字符 varchar(n) 固定长度, 最多65535个字符
tinytext 可变长度, 最多255个字符 text 可变长度, 最多65535个字符 mediumtext
可变长度, 最多2的24次方-1个字符 longtext 可变长度, 最多2的32次方-1个字符

MySql两种存储引擎的区别

MyISAM:

不支持事务, 但是每次查询都是原子的; 支持表级锁, 即每次操作是对整个表加锁; 存储表的总行数; 一个MYISAM表有三个文件: 索引文件、表结构文件、数据文件; 采用菲聚集索引, 索引文件的数据域存储指向数据文件的指针。辅索引与主索引基本一致, 但是辅索引不用保证唯一性。InnoDB:

支持ACID的事务, 支持事务的四种隔离级别; 支持行级锁及外键约束: 因此可以支持写并发; 不存储总行数; 一个InnoDB引擎存储在一个文件空间(共享表空间, 表大小不受操作系统控制, 一个表可能分布在多个文件里), 也有可能为多个(设置为独立表空, 表大小受操作系统文件大小限制, 一般为2G), 受操作系统文件大小的限制; 主键索引采用聚集索引(索引的数据域存储数据文件本身), 辅索引的数据域存储主键的值; 因此从辅索引查找数据, 需要先通过辅索引找到主键值, 再访问辅索引; 最好使用自增主键, 防止插入数据时, 为维持B+树结构, 文件的大调整。

三 InnoDB索引和MyISAM索引的区别

1 存储结构(主索引 / 辅助索引) InnoDB的数据文件本身就是主索引文件。而MyISAM的主索引和数据是分开的。

InnoDB的辅助索引data域存储相应记录主键的值而不是地址。而MyISAM的辅助索引和主索引没有多大区别。

innodb是聚簇索引, 数据挂在逐渐索引之下。

2 锁 MyISAM使用的是表锁

InnoDB使用行锁

3 事务 MyISAM没有事务支持和MVCC（多版本并发控制）

InnoDB支持事务和MVCC

4 全文索引 MyISAM支持FULLTEXT类型的全文索引

InnoDB不支持FULLTEXT类型的全文索引，但是InnoDB可以使用sphinx插件支持全文索引，并且效果更好

5 主键 MyISAM允许没有任何索引和主键的表存在，索引都是保存行的地址 InnoDB如果没有设定主键或非空唯一索引，就会自动生成一个6字节的主键，数据是主索引的一部分，附加索引保存的是主索引的值

6 外键 MyISAM不支持 InnoDB支持

```
create table activity_H5gusees_sonplist | CREATE TABLE
`activity_H5gusees_sonplist` (
  `songid` int(11) NOT NULL,
  `songurl` varchar(512) DEFAULT NULL,
  `songname` varchar(512) DEFAULT NULL,
  `artist` varchar(255) DEFAULT NULL,
  `level` tinyint(1) DEFAULT '1',
  `similarity_songname` varchar(20000) DEFAULT NULL,
  `create_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  `remark` varchar(512) DEFAULT NULL,
  PRIMARY KEY (`songid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

explain

显示了mysql如何使用索引来处理select语句以及连接表。可以帮助选择更好的索引和写出更优化的查询语句

```
mysql> explain select * from competition_coinsprize_record where
com_prizeid=20190724;
+----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | type | possible_keys | key | key_len | ref | rows | Extra |
+----+-----+-----+-----+-----+-----+-----+
```

```

-----+-----+-----+-----+-----+
| 1 | SIMPLE | competition_coinsprize_record | ref |
index_com_prizeid | index_com_prizeid | 4 | const | 11472 |
-----+-----+-----+-----+
-----+-----+-----+-----+

```

show processlist

查看连接情况

```

mysql> select client_ip,count(client_ip) as client_num from (select
substring_index(host,':',1) as client_ip from
information_schema.processlist ) as connect_info group by client_ip order
by client_num desc;

```

```

+-----+-----+
| client_ip | client_num |
+-----+-----+
| 10.10.14.24 | 8 |
| 10.10.14.22 | 7 |
| 10.10.14.23 | 3 |
| | 2 |
| 10.10.11.28 | 2 |
| 192.168.1.23 | 2 |
| 10.10.10.28 | 1 |
| 192.168.1.135 | 1 |
| 192.168.1.143 | 1 |
| 192.168.1.113 | 1 |
+-----+-----+

```

```

select * from information_schema.processlist where Command != 'Sleep'
order by Time desc limit 10;

```

sql查询

单表查询

```

select workid,userid,repotId from activity_work_soundreport where
userid=138392364;

```

交叉

```

select a.userid,a.workid,u.nickname from activity_work_soundreport as
a,user as u where a.userid=u.userid and userid=138392364;

```

子查询

```

select ws.userid,count(ws.songid) as num from(select * from

```

```
activity_work_soundreport where create_time>'2019-07-24' ) as ws group by  
userid order by num desc limit 10;
```

内连查询

```
SELECT c.id, title, content, finish_content, songids, duetids, gender,  
c.type, rank_type, minworkid, mp3name, pwd, show_field, owner, addtime,  
STATUS , begin_time, end_time, area, invalid, weight, pic2, pic,  
pic_check, video_check, picture_src, video_src, video_img, testers_check,  
testers, show_field, rule_id, rule_time, rule_requirement, rule_selection,  
rule_award, rule_remark, signup_id, remark_1, remark_2, remark_3,  
remark_4, remark_5, remark_6  
FROM competition AS c  
INNER JOIN competition_rule AS cr ON c.rule_id = cr.id  
AND c.invalid =0  
AND c.id = ".$competition_id."  
LEFT JOIN competition_signup AS cs ON c.signup_id = cs.id
```

左连接

left join on

右连接

right join on

增删改

```
insert into .....  
delete  
update
```

数据表添加修改字段建索引

```
```math  
ALTER TABLE tuikuan ADD column `finish_time` timestamp NULL DEFAULT
'0000-00-00 00:00:00' COMMENT '支付完成时间';
1 . ALTER TABLE
ALTER TABLE用来创建普通索引、UNIQUE索引或PRIMARY KEY索引。

ALTER TABLE table_name ADD INDEX index_name (column_list)

ALTER TABLE table_name ADD UNIQUE (column_list)

ALTER TABLE table_name ADD PRIMARY KEY (column_list)
```