

MySQL 使用规范

一、 表设计类

强制类规范

1. 创建表的存储引擎必须是 InnoDB, 是 MySQL 的数据库引擎之一 InnoDB 的最大特色就是支持了 ACID 兼容的事务 (Transaction) 功能。

2. 每个表必须显式的指定一个主键。

3. 不允许使用联合主键, 联合主键主要是解决一些重复值较多的列

4. 不允许使用外键, 表删除或备份等会涉及到外键关联表。

5. 不允许存在和主键重复的索引, 主键本身带索引。

6. 自增长字段必须是主键或唯一索引, 自增长字段本身不具有实际业务含义, 因为其自增所以建立唯一索引。

7. 不允许在数据库中存储诸如图片, 影像之类的二进制数据, 太占存储和检索慢, 网站使用独立文件图片服务器。

8. 不允许使用 TEXT 类型字段:

1. VARCHAR 到一定长度会自动转换成 TEXT 的

2. 能用 VARCHAR 搞定的, 绝对不用 TEXT, 否则性能要差很多

3. 对于文章内容等一类东西, 没得选择, 只能用 TEXT, 为此建议必须拆开存储

9. 建表时不允许显式的指定除了 utf8 之外的其他字符集。

10. 对于所有声明为 NOT NULL 的字段, 必须显式指定默认值。

11. 必须包含时间戳字段 DataChange_LastTime, 定义默认值为 CURRENT_TIMESTAMP 和 on update CURRENT_TIMESTAMP, 并添加索引。

建议类规范

1. 建议使用自增长字段作为主键。

2. 对较长的字符类型, 如果需要索引, 则建立前缀索引。

3. 不建议在数据库存放日志(错误日志、查询日志、慢查询日志、事务日志、二进制日志, 到 my.ini 或者你所使用的配置文件可修改日志路径

)。

4. 建议将字段都定义为 **not null**。
5. 选用能满足需求的最小类型,节省空间。
6. 避免使用保留字命名 **DB** 对象。
7. 对表和字段都添加备注说明。

二、 SQL 类

强制类规范

1. 禁止使用子查询。
2. 禁止使用 **select ***, 必须指定需要的字段。
3. **update/delete** 只能单表操作, 不允许多表关联, 不允许用子查询, 且一定要带 **where** 条件。
4. **insert** 语句要显式指定插入的列名, 且不允许使用 **insert select** 的形式。
5. 不允许使用存储过程、存储函数、触发器和视图。
6. 单条查询语句中, 不允许出现多于一次的 **join**。
7. 不要在 **where** 后的筛选字段上做运算。

建议类规范

1. 尽量不要在数据库里做运算。
2. 尽量不要做 ‘%’ 前缀模糊查询, 如 **like '%name'**。
3. 不要使用大偏移量的 **limit** 分页

如何优化 limit

当一个查询语句偏移量 **offset** 很大的时候, 如 **select * from table limit 10000,10**, 最好不要直接使用 **limit**, 而是先获取到 **offset** 的 **id** 后, 再直接使用 **limit size** 来获取数据。效果会好很多。

如:

```
select * From customers Where customer_id >=(  
select customer_id From customers Order By customer_id limit 10000,1  
) limit 10;。
```

4. 连接 MySQL 不要设置成 `autocommit=0`。
5. 批量 insert 语句最好采用 `bulk insert` 的方法，如 `insert into table(xxx) values (xxx),(xxx)`。
6. `update/delete` 尽量根据主键进行操作。
7. 尽量减少 `count()` 的使用，尤其是用来频繁获取全表记录数。
8. 使用 `group by` 时，如无排序的需求，建议加 `order by null`, 强制不排序。
9. `Join` 中使用的关联字段使用统一数据类型。