指导书

杭州海康威视数字技术股份有限公司

文档编号

# 技术开发组关系型SQL编写规范

密级级别：[内部公开]

生效时间： 单击此处输入日期。

保密期：无

杭州海康威视数字技术股份有限公司 版权所有

## 编写背景

组内开发人员在编写SQL时，由于对数据库引擎的内部实现理解不同，导致不同业务使用的SQL质量层次不齐，数据库难以维护和管理，数据库管理人员无法准确评估业务压力，进而影响线上服务质量，因此编写本SQL编写规范，统一SQL编写过程。

## 范围

本规范规定了SQL DQL和 DML 语言的编写总则，从书写格式和性能优化两方面归纳了 SQL 书写的具体要求，并给出 SQL 语句示例。

本规范适用于 技术开发组 内部所有开发和测试人员。

## 规范性引用文件

下列文件中的条款通过本规范的引用而成为本规范的条款，凡注明日期的引用文件，其随后所有的修改单（不包括勘误内容）或修订版均不适用于本规范，然而，鼓励根据本规范达成协议的各方研究是否可使用这些文件的最新版本。凡不注明日期的引用文件，其最新版本适用于本规范。

## 术语和定义

下列术语和定义适用于本规范

|  |  |
| --- | --- |
| 术 语 | 说 明 |
| 静态表 | 全字段为定长字段的表 |
| 小表 | 本文档所述”小表”是指关系型数据库中单表记录字段超过20个，单表记录数小于1000行的表，根据用户的具体使用场景，该约定值可能出现调整，以数据库管理人员的参考值为准。 |
| 大表 | 本文档所述”大表”是指关系型数据库中单表记录字段超过20个，单表记录数超过7800万行的表，根据用户的具体使用场景，该约定值可能出现调整，以数据库管理人员的参考值为准。 |
| 交叉表 | 被其他表引用的表。 |
| 驱动表 | 某些数据库引擎的解释器会按照从右到左的顺序处理FROM子句中的表名，因此FROM子句中写在最后的表将被最先处理。最右侧的表即为驱动表。 |

## SQL编写总则

#### 同一项目的SQL书写格式应该统一

#### SQL语句大小写应规范

说明：

1. SQL语句中的表名、表别名、字段名、序列等数据库对象应小写，以下SQL不符合该规范：
2. SQL语句中出现的系统保留字、内置函数名、SQL保留字、绑定变量等应大写

以下SQL不符合该规范：

应改为如下SQL：



#### SQL语句应具备良好的缩进和换行规范

说明：缩进应为1个Tab或4个字符，如果单行字符数过多，基于列对齐和纵向对齐原则，同层次SQL语句缩进应保持一致。应尽可能遵循以下规范：

1. SELECT/FROM/WHERE/ORDER BY/GROUP BY等子句应独占一行
2. SELECT/FROM子句内容如果只有一项，应与SELECT/FROM同占一行；如果多于一项，每一项都应独占一行，并在对应当前SELECT层级的基础上向右缩进1个Tab或4个字符
3. WHERE子句内容如果只有一项，应与WHERE同占一行；如果有多项，每一个条件应独占一行，并以AND开头，并在对应WHERE的基础上向右缩进1个Tab或4个字符
4. UPDATE/SET子句内容如果只有一项，应与SET同占一行；如果有多项，每一项应独占一行，并在对应SET的基础上向右缩进1个Tab或4个字符
5. INSERT子句如果有多项，左右括号以及每个表字段应独占一行，其中括号无缩进，表字段在对应括号的基础上向右缩进1个Tab或4个字符
6. VALUES子句如果有多项，左右括号以及每项的值应独占一行，其中括号无缩进，每项的值在对应括号的基础上向右缩进1个Tab或4个字符
7. SQL语句中不应出现空行
8. SQL语句内的算术运算符、逻辑运算符(AND、OR、NOT)、比较运算符(=、<=、>=、>、<、<>、BETWEEN AND)、IN、LIKE等运算符前后都应加一空格
9. SQL语句中逗号后应加一空格
10. 不等于应统一使用符号“<>”，避免使用“!=”

#### SQL语句注释应规范

说明：对于较复杂的SQL语句，为了增加可读性，应增加注释，并说明算法和功能。注释应单独成行，并放在语句前面。对过长的函数实现，应将语句按实现的功能分段加以概括性说明。对常量和变量注释时，应注释被保存值的含义，且应包括合法的取值范围说明。

#### SQL语句不应在客户端组织，而应在服务端组织

#### 【强制】SQL语句的语法应与所使用的数据库相适应

#### 【强制】超过三个表禁止join，需要join的字段，数据类型保持绝对一致；多表关联查询时，保证被关联的字段需要有索引

说明：即使双表join也要注意表索引，SQL性能。对于多表join，如果业务确实需要，则应确保该SQL的并发量非常低，且应确保条件过滤后参与join的子表数据量小于10000行。（）

建议：当业务开发人员发现，参与关联查询的子表数据超过限制阈值时，应分解关联查询，对每个要关联的表进行单表查询，然后将结果在应用程序中进行关联。



优化方案：对于每个要关联的表进行单表查询，然后将结果在应用程序中进行关联，SQL分解成以下查询来代替：



在多表关联查询WHERE条件中，需要明确写明每一个表的过滤条件，比如避免使用如下SQL：



在明确两个表都有相同time和type过滤条件情况下，建议修改为以下SQL：



#### 【强制】禁止跨DB的join语句

说明：跨DateBase的join语句，会增加模块间的耦合度，不利用数据库的业务拆分。

#### 【强制】禁止在业务的更新类SQL语句中使用join

以下SQL语句原则上禁止使用：



#### 表结构设计应尽可能满足数据库范式，但应考虑实际业务需要

说明：数据库范式指：第一范式。第二范式，第三范式，BC范式等。当完全满足第三范式时，数据字段没有任何冗余，但实际业务设计中，为了避免不必要的join，可以采用冗余部分字段，显著提高查询性能的方案优化。

#### 【强制】SELECT语句必须指定具体字段名称，不应该使用列的序号，禁止使用”\*”代替所有列名

返回结果字段的数量直接影响性能，造成磁盘和网卡压力，尤其在有Text，Varchar或Blob字段的情况下。



优化返回字段，将SQL改写为：



#### 【强制】INSERT语句应指定插入的字段名，不应在不指定字段名的情况下直接插入VALUES

以下SQL语句违反该项规范：



应修改为如下SQL：



#### 【强制】禁止对大表的全表扫描操作，禁止在不使用索引的情况下对大表进行操作

说明：大表的全表扫描，会导致数据库系统计算资源和内存的显著提高，同时会导致查询时延增加，为确保数据库系统稳定，必须确保业务SQL不触发大表全表扫描。

注意：由于不同数据库引擎的实现不同，是否触发大表的全表扫描，应根据用户实际使用的对应数据库引擎的分析结果。

#### 【强制】IN关键字的使用必须控制列表大小，不允许过大的IN关键字查询

说明：in关键字会触发数据库底层扫描，极度消耗计算资源，为了加速查询，应避免过大的列表做in查询，从而减轻数据库压力。

以下SQL语句违反该规范：



#### 【强制】禁止裸查询

说明：除静态表（单表记录数小于10000行）或小表，DML语句必须有WHERE条件，且使用索引查询。

#### 合理化利用数据库索引

说明：合理利用数据库索引可以加速查询效率，反之会导致内存占用率过高，同时降低数据写入性能。应遵循以下规范：

1. 索引的建立应慎重考虑，不是越多越好，索引可以提高相应的SELECT效率，但同时也会降低INSERT和UPDATE的效率
2. 查询列、排序列应与索引列次序保持一致
3. 禁止在WHERE子句中使用计算后的索引列
4. 禁止在WHERE子句索引列上使用函数或表达式，如果确实需要使用，则应建立对应的函数索引
5. （前提）禁止在WHERE子句中对索引列使用LIKE ‘%xxx%’和LIKE ‘%xxx’
6. 避免使用IS NOT NULL过滤条件
7. 禁止对索引列值进行隐式/显式转换
8. 禁止使用与索引列数据类型不一致的比较值
9. 对复合索引，WHERE子句中必须包含索引的第一列
10. IN和OR子句常会使用工作表，导致索引无效，如果不产生大量重复值，可以考虑把子句拆开，拆开的子句中应包含索引
11. 为保证SQL的执行效率，应避免使用UNION、OR子句，可考虑在应用中对结果集进行处理

#### 合理化使用数据引擎

说明：数据库引擎的数据表扫描会消耗计算资源，尤其在大表扫描时会消耗大量的计算资源，在SQL编写时应注意数据库引擎扫描的开销，避免过度扫表。SQL编写时应遵循以下规范：

1. 小表可使用全表扫描，当查询返回数据量超过表总数据的20%时允许使用全表扫描，其他情况下禁止使用全表扫描
2. 避免使用ORDER BY和GROUP BY等排序操作，如果业务需要在数据库层排序，应建立在索引列上，避免在非索引列排序
3. 避免使用UNION关键字，如果业务确实需要使用，应使用UNION ALL代替UNION
4. 避免使用HINT，HINT是用来强制SQL按照某个执行计划来执行，比如SQL\_NO\_CACHE，FORCE INDEX，IGNORE KEY，STRAIGHT JOIN等。如果业务确实需要，应与数据库管理人员协商，确保系统负载不会超过额定值

## 五. 修改记录

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **变更时间** | **变更人** | **变更说明** |
| 1 | 2020.01.10 | 江羽8 | 创建 |
|  |  |  |  |
|  |  |  |  |

1. 具体SQL整体
2. 规则分级
3. 黑名单功能
4. 日志解析和通告
5. Tidb Server降配，增加TiKV降配