|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ]草稿  [**√**]正式版本  [ ]正在修改 | 版权所有： | 福建智康云医疗科技有限公司 |
| 当前版本： | V1.0 |
| 修订日期： | 2019年08月05日 |

智康云

数据库设计规范

| **文档修订记录** | | | | | |
| --- | --- | --- | --- | --- | --- |
| 序号 | 版本号 | 版本日期 | 编辑人 | 复核人 | 修订内容 |
|  | V1.0 | 20190805 | 陈荔萍 | 陈荔萍 | 初稿（一律以PDF格式为标准形式，若您看到的此份文档不为PDF，则视为非正常途径获取，不具文档效力。） |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **文档类别、阅读范围** | | | | | |
| 编辑部门 | 智康云科技研发部 | 类别 | 规范文档 | 文档编号 | V1.0 |
| 阅读范围 | 1. 智康云科技研发部； | | | | |

目录

**[一、建表规约](#_Toc5860_WPSOffice_Level1)** **[2](#_Toc5860_WPSOffice_Level1)**

[（一）命名规范](#_Toc3116_WPSOffice_Level2) [2](#_Toc3116_WPSOffice_Level2)

[（二）设计规范](#_Toc16233_WPSOffice_Level2) [3](#_Toc16233_WPSOffice_Level2)

[（三）字段长度](#_Toc16422_WPSOffice_Level2) [4](#_Toc16422_WPSOffice_Level2)

**[二、索引规约](#_Toc3116_WPSOffice_Level1)** **[5](#_Toc3116_WPSOffice_Level1)**

**[三、SQL 语句](#_Toc16233_WPSOffice_Level1)** **[7](#_Toc16233_WPSOffice_Level1)**

**一、建表规约**

**（一）命名规范**

1.库名与应用名称尽量一致

2.表的命名最好是加上“业务名称\_表的作用”。

正例：alipay\_task / force\_project / trade\_config

3.表名、字段名必须使用小写字母或数字，禁止出现数字开头，禁止两个下划线中间只出现数字。

说明：MySQL 在 Windows 下不区分大小写，但在 Linux 下默认是区分大小写。因此数据库名、表名、字段名，都不允许出现任何大写字母，避免节外生枝。

正例：aliyun\_admin，rdc\_config，level3\_name

反例：AliyunAdmin，rdcConfig，level\_3\_name

3.表名不使用复数名词、动词时态只用一般现在时。

说明：表名应该仅仅表示表里面的实体内容，不应该表示实体数量，对应于 DO 类名也是单数形式，符合表达习惯。

4.禁用保留字，如 desc、range、match、delayed 等。

5.主键索引名为 pk\_字段名；唯一索引名为 uk\_字段名；普通索引名则为 idx\_字段名。

说明：pk\_ 即 primary key；uk\_ 即 unique key；idx\_ 即 index 的简称。

6.所有的临时表必须以tmp\_为前缀并以日期为后缀

说明：为清理临时表提供了方便

7.备份库，备份表必须以bak\_为前缀并以日期为后缀

说明：也是为了后面的清理提供了方便

1. 所有存储相同数据的列名和列类型必须一致（关联列）

9.有用到时分秒的字段名，一律以\_time结尾，只有年月日，一律以\_date结尾

**（二）设计规范**

1.所有表和字段都需要添加注释

说明：注释方法是使用comment从句添加表和列的备注，一般公司的是大型项目，为以后其他人的维护和更新提供了方便。

2.如果修改字段含义或对字段表示的状态追加时，需要及时更新字段注释。

3.字段允许适当冗余，以提高查询性能，但必须考虑数据一致。冗余字段应遵循： 不是频繁修改的字段；不是 varchar 超长字段，更不能是 text 字段。

正例：商品类目名称使用频率高，字段长度短，名称基本一成不变，可在相关联的表中冗余存储类目名称，避免关联查询。

4.单表行数超过 500 万行或者单表容量超过 2GB，才推荐进行分库分表。

说明：如果预计三年后的数据量根本达不到这个级别，请不要在创建表时就分库分表。

5.尽量做到冷热数据分离，减小表的宽度

说明：把经常一起使用的列放在一个表中，对select会更友好

6.禁止在表中建立预留字段

说明：无法确认存储的数据类型，无法选择合适的类型，并且以后修改的成本大于增加一个字段的成本，所以没有必要。

7.禁止在数据库中存储图片，文件等二进制数据

说明：一般图片和文件数据会很大，存储过多，会影响数据库的性能，解决办法：在专门的文件服务器上存储图片和文件，在数据库中存储对应的地址信息

8. 不得使用外键与级联，一切外键概念必须在应用层解决。

说明：外键与级联更新适用于单机低并发，不适合分布式、高并发集群；级联更新是强阻塞，存在数据库更新风暴的风险；外键影响数据库的插入速度。

9. 如果有全球化需要，所有的字符存储与表示，均以 utf-8 编码，注意字符统计函数的区别。

说明：

SELECT LENGTH("轻松工作")； 返回为 12

SELECT CHARACTER\_LENGTH("轻松工作")； 返回为 4

如果需要存储表情，那么选择 utfmb4 来进行存储，注意它与 utf-8 编码的区别。

10. 表达是与否概念的字段，必须使用 is\_xxx 的方式命名，数据类型是 unsigned tinyint（ 1 表示是，0 表示否）。

说明：任何字段如果为非负数，必须是 unsigned。

正例：表达逻辑删除的字段名 is\_delete，1 表示删除，0 表示未删除。

**（三）字段长度**

1. 合适的字符存储长度，不但节约数据库表空间、节约索引存储，更重要的是提升检索速度。

正例：如下表，其中无符号值可以避免误存负数，且扩大了表示范围。

对象 年龄区间 类型 字节 表示范围

人 150 岁之内 unsigned tinyint 1 无符号值：0 到 255

龟 数百岁 unsigned smallint 2 无符号值：0 到 65535

恐龙化石 数千万年 unsigned int 4 无符号值：0 到约 42.9 亿

太阳 约 50 亿年 unsigned bigint 8 无符号值：0 到约 10 的 19 次方

2.小数类型为 decimal，禁止使用 float 和 double。

说明：float 和 double 在存储的时候，存在精度损失的问题，很可能在值的比较时，得到不正确的结果。如果存储的数据范围超过 decimal 的范围，建议将数据拆成整数和小数分开存储。

3. 如果存储的字符串长度几乎相等，使用 char 定长字符串类型。

4. varchar 是可变长字符串，不预先分配存储空间，长度不要超过 5000，如果存储长度大于此值，定义字段类型为 text，独立出来一张表，用主键来对应，避免影响其它字段索引效率。

**二、索引规约**

1. 业务上具有唯一特性的字段，即使是多个字段的组合，也必须建成唯一索引。

说明：不要以为唯一索引影响了 insert 速度，这个速度损耗可以忽略，但提高查找速度是明显的；另外，即使在应用层做了非常完善的校验控制，只要没有唯一索引，根据墨菲定律，必然有脏数据产生。

2. 常见索引列建议：（在哪些列上建立索引）

①Select,update,delete语句的where从句中的列

②包含在order by,group by,distinct中的字段（联合索引）

③多表join的关联列

3. 超过三个表禁止join。需要 join 的字段，数据类型必须绝对一致；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。

4. 在 varchar 字段上建立索引时，必须指定索引长度，没必要对全字段建立索引，根据 实际文本区分度决定索引长度即可。

说明：索引的长度与区分度是一对矛盾体，一般对字符串类型数据，长度为 20 的索引，区分度会高达 90%以上，可以使用 count(distinct left(列名, 索引长度))/count(\*)的区分度 来确定。

5. 页面搜索严禁左模糊或者全模糊，如果需要请走搜索引擎来解决。

说明：索引文件具有 B-Tree 的最左前缀匹配特性，如果左边的值未确定，那么无法使用此索引。

6. 如果有 order by 的场景，请注意利用索引的有序性。order by 最后的字段是组合 索引的一部分，并且放在索引组合顺序的最后，避免出现 file\_sort 的情况，影响查询性能。

正例：where a=? and b=? order by c; 索引：a\_b\_c

反例：索引中有范围查找，那么索引有序性无法利用，如：WHERE a>10 ORDER BY b; 索引 a\_b 无法排序。

7. 利用覆盖索引来进行查询操作，避免回表。

说明：如果一本书需要知道第 11 章是什么标题，会翻开第 11 章对应的那一页吗？目录浏览一下就好，这个目录就是起到覆盖索引的作用。

正例：能够建立索引的种类：主键索引、唯一索引、普通索引，而覆盖索引是一种查询的一种效果，用 explain 的结果，extra 列会出现：using index。

8. 建组合索引的时候，区分度最高的在最左边。

正例：如果 where a=? and b=? ，a 列的几乎接近于唯一值，那么只需要单建 idx\_a 索引即可。

说明：存在非等号和等号混合判断条件时，在建索引时，请把等号条件的列前置。如：where a>? and b=? 那么即使 a 的区分度更高，也必须把 b 放在索引的最前列。 索引列的顺序（从左到右）：

①区分度最高的列放在联合索引的最左侧

②尽量把字段长度小的列放在联合索引的最左侧

③使用最频繁的列放在联合索引的最左侧

1. 避免建立冗余索引和重复索引

10.防止因字段类型不同造成的隐式转换，导致索引失效。

11. 创建索引时避免有如下极端误解：

1）宁滥勿缺。认为一个查询就需要建一个索引。

2）宁缺勿滥。认为索引会消耗空间、严重拖慢更新和新增速度。

3）抵制惟一索引。认为业务的惟一性一律需要在应用层通过“先查后插”方式解决。

**三、SQL 语句**

1. 不要使用 count(列名)或 count(常量)来替代 count(\*)，count(\*)是 SQL92 定义的标准统计行数的语法，跟数据库无关，跟 NULL 和非 NULL 无关。

说明：count(\*)会统计值为 NULL 的行，而 count(列名)不会统计此列为 NULL 值的行。

1. count(distinct col) 计算该列除 NULL 之外的不重复行数，注意 count(distinct col1, col2) 如果其中一列全为 NULL，那么即使另一列有不同的值，也返回为 0。

3. 当某一列的值全是 NULL 时，count(col)的返回结果为 0，但 sum(col)的返回结果为 NULL，因此使用 sum()时需注意 NPE 问题。

正例：可以使用如下方式来避免 sum 的 NPE 问题：SELECT IF(ISNULL(SUM(g)),0,SUM(g)) FROM table;

4. 使用 ISNULL()来判断是否为 NULL 值。

说明：NULL 与任何值的直接比较都为 NULL。

1） NULL<>NULL 的返回结果是 NULL，而不是 false。

2） NULL=NULL 的返回结果是 NULL，而不是 true。

3） NULL<>1 的返回结果是 NULL，而不是 true。

5.在代码中写分页查询逻辑时，若 count 为 0 应直接返回，避免执行后面的分页语句。

6.in 操作能避免则避免，若实在避免不了，需要仔细评估 in 后边的集合元素数量，控制在 1000 个之内。

7.充分利用表上已经存在的索引

说明：避免使用双%号的查询条件，如 a like ‘%123%’，使用left join或not exists来优化not in操作

1. 程序连接不同的数据库使用不同的账号，禁止跨库查询

9.禁止使用select \*必须使用select <字段列表>查询

说明：：使用select \*会消耗更多的CPU和IO以及网络带宽资源，无法使用覆盖索引。而且可减少表结构变更带来的影响

10.禁止使用不含字段列表的insert语句

说明：Insert into t values(‘a’,’b’,’c’)应替换为Insert into t(c1,c2,c3) values(‘a’,’b’,’c’);减少表结构变更带来的影响

11.避免使用子查询，可以把子查询优化为join操作

说明：子查询的结果集无法使用索引，会消耗过多的CPU及IO资源

12.避免使用join关联太多的表（建议不超过5个）

13. 减少同数据库的交互次数

说明：数据库更适合处理批量操作，合并多个相同的操作到一起，可以提高处理效率

14. 禁止使用order by rand()进行随机排序

说明：order by rand()的操作是会把表中所有符合条件的数据装载到内存中，并分配一个随机值，进行排序，这样会消耗大量的CPU和IO及内存资源，解决办法：推荐在程序中获取一个随机值，然后从数据库中获取数据的方式

15. Where从句中禁止对列进行函数转换和计算

说明：进行函数转换和计算则无法使用索引，例where date(createtime)=’20160901’可转换为where createtime>= ‘20160901’ and createtime< ‘20160902’

16. 在明显不会有重复值时使用UNION ALL而不是UNION

说明：UNION会把所有数据放到临时表中后再进行去重操作，UNION ALL不会再对结果集进行去重操作

17. 拆分复杂的大SQL为多个小SQL

说明：在MYSQL一个SQL只能使用一个CPU进行计算,SQL拆分后可以通过并行执行来提高处理效率

**四、PACS字段**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文名 | 字段名 | 字段类型 | 长度 | 描述 |
| 医院名称 | hospital\_name | varchar | 32 |  |
| 医院编码 | hospital\_id | varchar | 32 |  |
| 病例号 | medi\_id | varchar | 32 |  |
| HIS申请单号 | his\_request\_id | varchar | 32 |  |
| 检查号 | examine\_id | varchar | 32 |  |
| 检查编号 | accession\_number | varchar | 32 |  |
| 患者编号 | patient\_id | varchar | 32 |  |
| 住院号 | in\_patient\_id | varchar | 32 |  |
| 床号 | bed\_id | varchar | 32 |  |
| 患者姓名 | patient\_name | varchar | 32 |  |
| 英文名 | english\_name | varchar | 32 |  |
| 患者性别 | patient\_sex | varchar | 8 |  |
| 患者年龄 | patient\_age | varchar | 8 |  |
| 患者年龄 | patient\_age\_unit | varchar | 8 | 岁、月、天 |
| 患者生日 | patient\_birthdate | varchar | 32 |  |
| 患者住址 | patient\_address | varchar | 256 |  |
| 手机号码 | mobile\_phone | varchar | 32 |  |
| 身份证号 | id\_card\_numbe | varchar | 32 |  |
| 检查类型 | modality | varchar | 32 |  |
| 检查部位 | examine\_body | varchar | 256 |  |
| 就诊类别 | diagnosis\_type | varchar | 32 |  |
| 记录时间 | record\_time | varchar | 32 |  |
| 申请科室 | request\_dept | varchar | 32 |  |
| 申请医生 | request\_doc | varchar | 32 |  |
| 临床诊断 | cinical\_diagnosis | varchar | 32 |  |
| 患者病史 | patinet\_history | varchar | 32 |  |
| 报告时间 | report\_time | varchar | 32 |  |
| 报告医生 | report\_doc | varchar | 32 |  |
| 审核时间 | audit\_time | varchar | 32 |  |
| 审核医生 | audit\_doc | varchar | 32 |  |
| 影像所见 | image\_evidences | varchar | 2048 |  |
| 影像诊断 | image\_diagnosis | varchar | 2048 |  |
| 阳性标记 | is\_masculine | varchar | 32 |  |
| 报告状态 | report\_status | varchar | 32 |  |
| 标记时间 | flag\_time | varchar | 32 |  |
| 报告状态码 | report\_state | varchar | 32 |  |
| 注册时间 | register\_time | varchar | 32 |  |
| 修订时间 | revise\_time | varchar | 32 |  |
| 修订医生 | revise\_doc | varchar | 32 |  |
| 创建时间 | create\_time | datetime |  |  |
| 修改时间 | modify\_time | datetime |  |
| 接收时间 | receive\_time | datetime |  |
| 节点描述 | description | varchar | 16 |  |
| 节点地址 | ip | varchar | 16 | 192.168.168.168暂不考虑IPV6 |
| 节点标识 | ae | varchar | 16 |  |
| 节点端口 | port | int |  | 1-65535 |
| 重试次数 | retry\_count | tinyint |  |  |
| 重试间隔（秒） | retry\_interval | tinyint |  |  |
| Dicom传输语法 | transfer\_syntax | varchar | 128 |  |
| 是否支持存储服务 | sup\_store\_services | bit |  |  |
| 是否支持存储确认 | sup\_store\_commitment | bit |  |  |
| 是否支持查询获取 | sup\_query\_retrieve | bit |  |  |
| 是否支持工作清单 | sup\_worklist | bit |  |  |
| 超时时间（秒） | time\_out | tinyint |  |  |
| 是否已注册 | Is\_registed | bit |  |  |
| 其他患者编号 | other\_patient\_id | varchar | 32 |  |
| 其他患者姓名 | other\_patient\_name | varchar | 32 |  |
| 检查数量 | studies\_count | int |  |  |
| 序列数量 | series\_count | int |  |  |
| 影像数量 | images\_count | int |  |  |
| 影像是否在线 | is\_online | bit |  |  |
| 接收时间 | receive\_time | datetime |  |  |
| 检查日期 | study\_date | varchar | 16 |  |
| 检查时间 | study\_time | varchar | 16 |  |
| 检查描述 | study\_description | varchar | 16 |  |
| 咨询医师姓名 | referring\_physicians\_name | varchar | 64 |  |
| 执行医师姓名 | performing\_physicians\_name | varchar | 64 |  |
| 检查名称 | name\_of\_physicians\_reading\_study | varchar | 64 |  |
| 检查ID | study\_id | varchar | 32 |  |
| 检查实例UID | study\_instance\_uid | varchar | 64 |  |
| 媒体文件集ID | media\_file\_set\_id | varchar | 16 |  |
| 媒体文件集UID | media\_file\_set\_uid | varchar | 64 |  |
| 序列号码 | series\_number | varchar | 16 |  |
| 序列实例UID | series\_instance\_uid | varchar | 64 |  |
| 序列日期 | series\_date | varchar | 16 |  |
| 序列时间 | series\_time | varchar | 16 |  |
| 序列描述 | series\_description | varchar | 64 |  |
| 传输语法UID | transfer\_syntax\_uid | varchar | 64 |  |
| SOP类UID | sop\_class\_uid | varchar | 64 |  |
| SOP实例UID | sop\_instance\_uid | varchar | 64 |  |
| 影像号 | image\_number | varchar | 16 |  |
| 影像日期 | image\_date | varchar | 16 |  |
| 影像时间 | image\_time | varchar | 16 |  |
| 影像行数 | image\_rows | int |  |  |
| 影像列数 | image\_columns | int |  |  |
| 比特分配 | bits\_allocated | int |  |  |
| 比特存储 | bits\_stored | int |  |  |
| 影像度量标准 | photo\_metric | varchar | 16 |  |
| 层面厚度 | slice\_thickness | varchar | 32 |  |
| 片间距 | spacing\_between\_slices | varchar | 32 |  |
| 本地影像文件路径 | image\_file\_name | varchar | 128 |  |
| 影像文件大小 | image\_file\_size | int |  |  |
| CRC值 | crc\_value | int |  |  |
| 影像主键 | image\_key | bigint |  |  |
| 患者主键 | patient\_key | bigint |  |  |
| 检查主键 | study\_key | bigint |  |  |
| 序列主键 | series\_key | bigint |  |  |