贵州智慧通酒店有限公司

密级：保密

酒店帮数据库设计标准规范

**（V1.3.2）**

**贵州智慧通酒店信息科技有限公司**

版本修订记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本号** | **版本日期** | **修订者** | **修订状态** | **修订内容** |
| V0.1 | 2018-09-27 | 黄金翔 | C | 创建初始版本文件。 |
| V0.2 | 2018-10-16 | 黄金翔 | M | 完善相关内容，完成第一版 |
| V0.3 | 2018-10-18 | 黄金翔 | M | 新增、修改4.6章节字段规范内容 |
| V1.3.2 | 2018-12-04 | 黄金翔 | M | 修改在研发过程中调整的标准 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

修订状态：C – 创建 M – 修改 D – 删除 R – 发布

目 录

[1 引言 5](#_Toc531705657)

[1.1 背景 5](#_Toc531705658)

[1.2 目的 5](#_Toc531705659)

[1.3 名词解释 5](#_Toc531705660)

[2 数据库设计原则 6](#_Toc531705661)

[3 数据库设计概要 6](#_Toc531705662)

[4 总体设计 7](#_Toc531705663)

[4.1 部署设备 7](#_Toc531705664)

[4.2 MYSQL参数设置 9](#_Toc531705665)

[4.3 存储规划 11](#_Toc531705666)

[4.4 数据库表空间设计 11](#_Toc531705667)

[4.4.1 设计原则 11](#_Toc531705668)

[4.4.2 空间规划 12](#_Toc531705669)

[4.5 用户权限管理设计 12](#_Toc531705670)

[4.5.1 设计原则 12](#_Toc531705671)

[4.5.2 用户规划 13](#_Toc531705672)

[4.6 数据表结构及规范设计 13](#_Toc531705673)

[4.6.1 实体对象命名规范 13](#_Toc531705674)

[4.6.2 字段命名规范 15](#_Toc531705675)

[4.7 数据表分区设计 16](#_Toc531705676)

[4.7.1 原则 16](#_Toc531705677)

[4.7.2 分区规划 17](#_Toc531705678)

[4.8 索引优化设计 18](#_Toc531705679)

[4.8.1 原则 18](#_Toc531705680)

[4.8.2 规划 19](#_Toc531705681)

# 引言

## 背景

略。

## 目的

本文档旨在描述酒店帮综合服务平台数据库存储及系统规划的方案和建议，该文档用于指导酒店帮MySql数据库的数据库设计和标准规范。

本文预期的读者为：

* 酒店帮项目组领导
* 酒店帮数据组成员

## 名词解释

**酒店帮综合服务平台**：集成相关应用系统、数据库形成对外提供的服务平台

**酒店帮商户助手**：提供酒店自助终端，退款管、对账管理、OTA管理等。

**chais：**酒店帮信息管理系统（全称china hotel assist information system）

**chbis：**酒店商户信息管理系统（全称china hotel business information system）

**chaisdb：**酒店帮信息管理系统数据库（酒店帮运维云）

**chbisdb：**酒店商户信息管理系统数据库（酒店帮运营云）

**酒店帮自助终端：**完成自动办理入住、退房的自助终端机

# 数据库设计原则

* 高可用性原则。根据OLAP和OLTP系统特点，酒店帮自助终端机实时产生数据定为OLTP系统，数据上传至chaisdb，由酒店帮信息管理系统进行管理；为保证系统性能，构建高可用性数据库系统，后期规划部署主从数据集群，并做读写分离，分库分表操作。
* 扩展性原则。结合数据存储、增长现状，预估数据存储空间需求。
* 安全性原则。对用户、角色进行权限控制；重要数据定期备份。
* 关注重点原则。 针对具体数据对象的访问频度及性能要求，结合主机、存储、分区等做好数据库性能设计。

# 数据库设计概要

目前chaisdb和chbisdb仅仅做为《酒店帮综合服务平台技术规范与总体建设方案 - V1.0》中架构设计其中数据存储。

* 应用方面：当前还未有对应的应用进行使用。
* 设计方面：当前数据库整体设计单节点进行应用，后期随着数据量增大会考虑进行集群环境的搭建。
* 管理方面：当前仅仅建立测试用户chais，拥有DBA权限。为开发做测试使用。后期会将权限收回，仅仅给与connect、和resource相关权限。

当前数据库存储主要分为两类，一类为chaisdb酒店帮运维数据，包括：酒店管理、终端管理、日志管理、告警管理、设备管理、事件管理、配置管理、运维服务管理、授权管理等。另一类为chbisdb酒店帮云营数据，该数据库存放自助终端上传过来的订单数据、账务数据、会员数据、房间/房型数据等。同时，基于这些业务数据对数据进行清洗、整合、加工，后期形成专题数据进入大数据平台做数据分析、数据挖掘工作，支持各类应用的访问。

# 总体设计

## 部署设备

目前chaisdb和chbisdb部署于X86服务器上，具体的配置如表4-1所示：

表格 4‑1部署环境配置

|  |  |  |  |
| --- | --- | --- | --- |
| **分类** | **参数** | **值** | **备注** |
| 华为云服务器配置 | 机器型号 | 华为云服务器 |  |
| CPU | 物理CPU 1个，每颗2核4线程，主频2.2G； |  |
| 内存 | 8G |  |
| 内置硬盘 | 一个40G的内置硬盘，挂载了一个500G硬盘 |  |
| 硬盘分区 | 做了200G逻辑分区，空闲分区300G |  |
| 操作系统 | CentOS Linux release 7.4.1708 64位 |  |
| 数据库 | 数据库版本 | MySql 8.0.1.2( 64位) |  |
| 网络配置 | IP地址 | 139.159.140.8 |  |
| 网关 |  |  |
| DNS |  |  |
| 带宽 | 5M |  |
| 带宽共享类型 | 独享 |  |

当前华为云服务器开放的端口如下表所示：

表格 网络端口配置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方向** | **类型** | **协议** | **端口范围/ICMP类型** | **远端** |
| 出方向 | IPv4 | Any | Any | Any |
| 入方向 | IPv4 | Any | Any | sg1529494454873 (7a814832-13ec-437c-9490-f20ca1849e3d) |
| 入方向 | IPv4 | TCP | 22 | 0.0.0.0/0 |
| 入方向 | IPv4 | TCP | 3389 | 0.0.0.0/0 |
| 入方向 | IPv4 | TCP | 3389 | 0.0.0.0/0 |
| 入方向 | IPv4 | TCP | 8929 | 0.0.0.0/0 |
| 出方向 | IPv4 | TCP | 8929 | 0.0.0.0/0 |
| 入方向 | IPv4 | TCP | 30401 | 0.0.0.0/0 |
| 出方向 | IPv4 | TCP | 30401 | 0.0.0.0/0 |
| 入方向 | IPv4 | TCP | 30402 | 0.0.0.0/0 |
| 出方向 | IPv4 | TCP | 30402 | 0.0.0.0/0 |
| 入方向 | IPv4 | TCP | 30403 | 0.0.0.0/0 |
| 出方向 | IPv4 | TCP | 30403 | 0.0.0.0/0 |
| 入方向 | IPv4 | TCP | 3306 | 0.0.0.0/0 |
| 入方向 | IPv4 | TCP | 33060 | 0.0.0.0/0 |

基于以上配置，当前将数据库服务器作为单节点实例，主要使用端口为3306、33060。酒店帮数据库实例，用于存放从自助终端机上传的数据，以及PMS、银联、税控、公安等各类专题应用数据。

## MYSQL参数设置

MySql数据库参数设置如表4-2所示。

表格 4‑2（a）chais实例系统参数

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **参数** | **建议值** | **依据** |
| 1 | server\_id | 1 | 指定服务器ID |
| 2 | port | 3306 | MySQL默认端口 |
| 3 | basedir | /data/servers/mysql | MySQL安装基目录的路径 |
| 4 | datadir | /data/mysql/data | MySQL数据文件存放路径 |
| 5 | lower\_case\_table\_names | on | 数据库是否区分大小写 |
| 6 | character\_set\_server | utf8 | 数据库字符集 |
| 7 | innodb\_undo\_directory | /data/mysql/undo | Undo路径 |
| 8 | innodb\_undo\_tablespaces | 4 | 设置4组undo |
| 9 | innodb\_page\_size | 16k | 指定页面大小 的InnoDB 表空间 |
| 10 | open\_files\_limit | 10240 | 允许数据库打开的文件数 |
| 11 | back\_log | 600 | MySQL可以拥有的未完成连接请求的数量 |
| 12 | max\_connections | 3000 | MySQL数据库最大连接数 |
| 13 | max\_connect\_errors | 6000 | MySQL可阻断错误连接数的数量 |
| 14 | table\_open\_cache | 1000 | 可缓存表数据量 |
| 15 | innodb\_buffer\_pool\_size | 2048M | InnoDB缓存表和索引数据的内存区域 |
| 16 | innodb\_data\_file\_path | ibdata1:512M:autoextend | 默认表空间增长控制 |
| 17 | innodb\_log\_buffer\_size | 16M | InnoDB 用于写入磁盘上日志文件 的缓冲区的大小 |
| 18 | innodb\_lock\_wait\_timeout | 120 | 放弃之前InnoDB 事务等待行锁定的时间长度 |
| 19 | innodb\_file\_per\_table | on | ，InnoDB存储数据和索引在一个单独的每个新创建的表 .ibd 文件 |
| 20 | log\_error | /data/mysql/log/error.log | 默认错误日志记录 |
| 21 | pid\_file | /data/mysql/mysqld.pid | 进程标识文件的路径名 |
| 22 | …… |  |  |

## 存储规划

目前酒店帮数据库系统使用的是华为云服务器内置40G大小硬盘，挂载500G大小的外部磁盘。已分配200G空间用于MySql数据库的数据存储，剩余300G大小的存储空间。暂时无法评估空间增长量，后期根据终端机每天产生的数据量作为参考依据，在规划空间使用量。

数据库软件目录如下：

|  |  |
| --- | --- |
| **安装路径** | **作用** |
| /data/mysql/tmp | 存放数据库启动时的临时行文件 如：mysql.sock |
| /data/mysql/data | 存放数据库数据文件和日志文件，其中redo日志文件设置3组，每组128M 用于数据回滚，恢复操作。 |
| /data/mysql/dumps | 存放数据库备份文件 |
| /data/mysql/log | 存放数据库操作日志、错误日志文件等 |
| /data/mysql/redo | 存放数据库的redo日志 |
| /data/mysql/undo | 存放MySql数据库undo文件，设置为4组undo |
| /data/servers/mysql | 存放数据库安装文件 |

## 数据库表空间设计

### 设计原则

原则上每个schema对应一个表空间和一个相应索引表空间；但根据MySql数据库官方建议，用户和表空间是进行区分的。通过innodb\_file\_per\_table该参数行驶MySsql独有的表空间管理特性。每个Table有单独对应的表空间进行使用。大数据量的表对应一个单独的表空间和自己的索引表空间。索引表空间大小依据实际情况而定，一般为数据表空间大小的50%-70%。

表空间的设计需从以下几方面考虑：

* 控制用户所占用的表空间配额，防止硬盘空间耗竭
* 控制数据库所占用的磁盘空间，避免因为磁盘空间的耗竭而导致其它应用服务的崩溃
* 灵活放置表空间，提高数据库的输入输出性能，利于数据的备份与恢复等管理工作
* 日志文件与数据文件分开放，提高数据库安全性。

### 空间规划

酒店帮数据库实例主要用于存储从酒店帮自助终端机上传的数据。该实例通过使用MySql数据库innodb\_file\_per\_table该参数进行管理，确定每个表有自己独立表空间且表上的索引数据也是存放在自己独立的表空间上，随着后期业务与架构变更可适当的合并对应的表空间做相关调整。

## 用户权限管理设计

### 设计原则

总体遵循运营数据与运维数据进行分离的原则。运维数据库用户设计遵循生产、测试、查询分离的原则；运营数据库用户设计以系统为单位，每个系统独立一个用户。

数据库具体用户设计从以下几方面考虑：

* 用户名规范：数据库用户名以数据库名的前缀，如：chais为前缀，数据应用数据命名为：chais\_+意义名称。
* 密码设置：密码设置不允许跟用户名相同，密码长度需大于用户名字符长度
* 权限控制：权限分配需至少控制到表级
* 表空间设置：每个Table用独立的表空间

### 用户规划

基于现有chaisdb用户共3个。具体如表3-1所示。

表格 3‑1用户清单

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **用户名** | **系统** | **用途** |
| 1 | chais | 自助终端机、酒店帮信息系统 | 用于生产系统、研发测试。仅有DML权限 |
| 2. | chbis | 酒店帮商户助手 | 用于终端业务数据上传的用户 |
| 2 | chais\_admin |  | MySql生产库DBA用户，管理用户授权、DDL、DML管理等。 |
| 3 | chais\_dm | 自助终端机、酒店帮信息系统 | 存放标准代码、业务系统代码、行政区划等 |
| 4 | …… | …… | …… |

## 数据表结构及规范设计

### 实体对象命名规范

命名规则遵循如下原则：实体类型分类标识 +数据功能分类标识+意义名词，中间以“\_”隔开。例如：自助终端信息表（命名为：cf\_ti\_terminal）,自助终端参数配置信息表（命名为：cf\_pt\_config）。

|  |  |  |
| --- | --- | --- |
| 实体类型分类约定 | | |
| 名词 | **标识** | **说明** |
| 业务表 | T | 所有业务数据表 |
| 代码表 | DM | 代码表 |
| 日志表 | LOG | 日志表 |
| 临时表 | TMP | Temporary |
| 配置表 | CF | 终端机上参数配置信息存放 |
| 系统表 | SYS | 用户、权限、功能等配置信息 |
| 普通视图 | VW |  |
| 主键 | PK |  |
| 外键 | FK |  |
| 触发器 | TRG |  |
| 存储过程 | SP | Stored Procedure |
| 函数 | FUN | Function |
| 序列 | SEQ | Sequence |

|  |  |  |
| --- | --- | --- |
| 数据功能分类标识约定 | | |
| 名词 | **标识** | **说明** |
| 自助终端信息 | TI | 自助终端信息相关数据 |
| PMS管理系统 | PMS | pms获取相关数据 |
| 参数信息 | PT | 终端机上参数信息 |
| …… |  |  |

### 字段命名规范

命名规则遵循如下原则：字段标识分类+意义名词，中间以“\_”隔开。例如：酒店ID（命名为：hi\_id）,自助终端ID（命名为：ti\_id）。

* 字段意义名词采用行业命名方式，参考别样红、绿云等PMS厂商命名规范，结合自身规范进行命名。
* 字段类型尽可能采用字符型，除主键、非空字段、表码字段、时间字段、金额字段外，其他字段尽可能采用可变长字符类型。
* 记录时间的字段采用data，“日期型”为格式为：yyyy-mm-dd hh24:mi:ss

|  |  |  |
| --- | --- | --- |
| 字段标识分类 | | |
| 名词 | **标识** | **说明** |
| 集团 | BLOC |  |
| 酒店 | HI |  |
| 自助终端信息 | TI |  |
| 订单 | OR |  |
| 消费品 | CON |  |
| 房型/房间 | RO |  |
| 入住人 | CI |  |
| 硬件版本 | HV |  |
| 软件版本 | SV |  |
| 硬件 | HW |  |
| 创建 | C |  |
| 修改 | M |  |
| 参数 | P |  |
| 账务 | BILL |  |
| 支付 | PAY |  |
| …… |  |  |

## 数据表分区设计

### 原则

基于查询性能的考虑，原则上超过500万的数据表需考虑分区存放。分区类型主要考虑列表分区（按地市行政区划），范围分区（按月/旬/季），散列分区。

### 分区规划

#### 列表分区

对于分地市抽取的大数据量表，需考虑按地区码进行数据分区，单个地市数据量超过1000万，考虑建立子分区，具体如表4-10所示。

表格 4‑10列表分区

|  |  |  |
| --- | --- | --- |
| **序号** | **表名** | **分区方法** |
|  | 贵州 | 按省级行政区划分区 |
|  | 湖北 | 按省级行政区划分区 |
|  | 湖南 | 按省级行政区划分区 |
|  | 重庆 | 按省级行政区划分区 |
|  | 广西 | 按省级行政区划分区 |
|  | 广东 | 按省级行政区划分区 |
|  | 浙江 | 按省级行政区划分区 |

#### 范围分区

对于每日增量较多的大数据量表，需考虑按时间范围（按月/旬/季）进行数据分区存储，如表4-11所示。

表格 4‑11范围分区

|  |  |  |
| --- | --- | --- |
| **序号** | **表名** | **分区方法** |
|  | 自助终端机订单数据 | 以订单时间为划分依据，每月一个数据分区 |
|  | 退房记录数据 | 以退房记录时间为划分依据，，每月一个数据分区 |
|  | 入住人员数据 | 以入住时间为划分依据，每月一个数据分区 |
|  | 账务数据 | 以上传时间为划分依据，每月一个数据分区 |

#### 散列分区

对于无法进行列表分区、范围分区的大数据量表，需要根据其实际数据值建立hash分区，如人要素表，具体参见表4-12所示。

表格 4‑12散列分区

|  |  |  |
| --- | --- | --- |
| **序号** | **表名** | **分区方法** |
| 1 | 房型记录数据 | 以房型编号为hash分区 |
| 2 | 消费品记录数据 | 以消费品编号为hash分区 |

## 索引优化设计

### 原则

索引的设计需从以下几方面考虑：

* 选择合适的索引。原则上不使用位图索引，采用B\*Tree索引
* 单次检索数据量低于全表20%，需考虑建立索引，否则不推荐
* 小于10000条记录的数据表不宜建索引
* 经常使用的大数据量表需考虑建立字段级索引，如订单数据
* 多表连接关联字段需考虑建立索引
* 常用多条件查询需考虑建立组合索引

### 规划

需考虑建立索引的数据字段为：

* 身份证及其他证件号码
* 订单编号、终端机编号
* 关联条件字段
* 数据表主键字段
* 时间戳字段