UOKO技术研发中心

MySQL数据库设计规范

优客逸家（成都）信息科技有限公司

2019年3月26日

# 目录

目录

[目录 2](#_Toc4500568)

[1.概述 4](#_Toc4500569)

[1.1 背景 4](#_Toc4500570)

[1.2 适用范围 4](#_Toc4500571)

[1.3 术语及定义 4](#_Toc4500572)

[2.数据库策略 5](#_Toc4500573)

[2.1 总则 5](#_Toc4500574)

[2.2 设计策略 5](#_Toc4500575)

[2.3 对象长度策略 5](#_Toc4500576)

[2.4 数据完整性策略 5](#_Toc4500577)

[2.5 设计与性能的权衡策略 5](#_Toc4500578)

[3 基本规范 6](#_Toc4500579)

[3.1 功能 6](#_Toc4500580)

[3.2 对象使用 6](#_Toc4500581)

[3.3 设计规范 6](#_Toc4500582)

[4.对象命名规范 7](#_Toc4500583)

[4.1 总则 7](#_Toc4500584)

[4.2 库命名 7](#_Toc4500585)

[4.3 表命名 7](#_Toc4500586)

[4.4 字段命名 8](#_Toc4500587)

[4.5 索引命名 8](#_Toc4500588)

[4.6 用户 9](#_Toc4500589)

[5.对象设计规范 9](#_Toc4500590)

[5.1 总则 9](#_Toc4500591)

[5.2 库设计规范 9](#_Toc4500592)

[5.3 表设计规范 9](#_Toc4500593)

[5.4 字段设计规范 10](#_Toc4500594)

[5.5 索引设计规范 10](#_Toc4500595)

[6.SQL开发规范 11](#_Toc4500596)

[6.1 总则 11](#_Toc4500597)

[6.2 DML 11](#_Toc4500598)

[6.3 DDL 12](#_Toc4500599)

[6.4 DQL 12](#_Toc4500600)

[6.5 条件 12](#_Toc4500601)

[7.行为规范 13](#_Toc4500602)

[7.1 建表申请 13](#_Toc4500603)

[7.2 重大操作 13](#_Toc4500604)

[7.3 危险行为 13](#_Toc4500605)

[8.数据库脚本 14](#_Toc4500606)

[8.1 建表脚本 14](#_Toc4500607)

[9.附录 15](#_Toc4500608)

[9.1 MySQL保留关键字 15](#_Toc4500609)

[9.2 字段类型策略 19](#_Toc4500610)

[9.3 数据类型占位对照表 19](#_Toc4500611)

# 1.概述

## 1.1 背景

针对MySQL关系型数据库的特性，拟定业务开发过程中的数据库设计规范，旨在通过本规范的建议和约束，培养开发人员统一的编码风格和良好的编码习惯，提高程序的可靠性、可读性、可维护性和一致性等，保证MySQL数据库的健壮性。

## 1.2 适用范围

本规范适用于所有MySQL相关业务的开发人员、数据库开发DBA及运维DBA。

## 1.3 术语及定义

本规范采用以下的术语描述：

* 禁止：编码明令禁止使用的原则。
* 规则：编码强制遵守的原则。
* 说明：对此规则进行必要的解释。
* 示例：对此规则或建议给出示例。

# 2.数据库策略

## 2.1 总则

* 数据库设计遵循关系数据库的范式理论。
* 数据库建模要基于统一的元数据管理机制。
* 数据模型全局单一，所有公共的数据模型得到共享。
* OLTP与OLAP系统分开设计。

## 2.2 设计策略

规范数据库设计的主要目的，是为了满足当前业务及扩展的需求。主要考虑如下方面：

* 方便业务功能实现和业务功能扩展。
* 方便设计开发、增强系统的稳定性和可维护性。
* 保证数据完整性和准确性。
* 提高数据存储效率，在满足业务需求的前提下，使时间开销和空间开销达到优化平衡。

## 2.3 对象长度策略

数据库字段的长度要考虑业务对象的类型、数据库所用字符集、时间格式来设计相对准确的长度，满足业务需要，同时保证数据库的性能，避免不必要的开销。

## 2.4 数据完整性策略

必须遵循数据库设计的第二范式，根据业务需要尽量满足第三范式。

**数据完整性尽量通过业务逻辑实现，数据库设计禁止使用存储过程、视图、事件、触发器、外键等特性。**

## 2.5 设计与性能的权衡策略

数据的标准化有助于消除数据库中的数据冗余。如果数据冗余低，数据的一致性容易得到保证，如无特殊理由，OLTP系统的设计应当遵循第三范式，对于OLAP系统，为了减少表间连接查询的操作，提高系统的响应时间，合理的数据冗余是必要的。

# 3 基本规范

数据库的主要目的是存储，为了达到高效使用，应该禁止部分功能和规范对象使用。

## 3.1 功能

【禁止】禁用存储过程

【禁止】禁用视图

【禁止】禁用事件

【禁止】禁用触发器

【禁止】禁用外键

【禁止】禁用分区表

【禁止】禁用保留字命名

【禁止】禁用ENUM、SET类型

【禁止】禁用TEXT/BLOB类型

## 3.2 对象使用

【禁止】禁用“-”中划线命令

【禁止】禁用select \*

【禁止】禁用频繁短连接

【禁止】禁用数据库层运算

【禁止】禁止滥用外连接

【禁止】禁止超过三个表的关联

【禁止】禁用负责SQL、大事务、大批量

【禁止】禁用无条件或永真条件的DELETE和UPDATE语句

## 3.3 设计规范

【禁止】禁止在线上做数据库压力测试

【禁止】禁止从测试、开发环境直连线上数据库

【禁止】禁止在数据库中存储图片、文件等大对象

【禁止】禁止在数据库中存储明文密码

# 4.对象命名规范

## 4.1 总则

【规则】特殊约定或缩写，要有注释说明

【规则】采用英文单词、英文短语作为名称

【规则】库、表、字段命名的字符范围为：a-z, 0-9 和\_(下划线)

【规则】长度控制在18个字符以内，缩写要易于理解，符合通用的习惯

## 4.2 库命名

【规则】单库仅使用a-z，分库名称为“库名\_编号”

【规则】命名规则如下：项目简称+1位数据库类型代码+识别代码+序号

* 只有一个数据库，则不加序号，否则末尾增加序号。
* 生产库则不加识别代码，否则需要增加识别代码DEV或TEST
* 如果只作历史库，则只需要项目简称+H+序号

【说明】数据库代码：

|  |  |  |  |
| --- | --- | --- | --- |
| **类型代码** | | **识别代码** | |
| T | 业务型数据库 | DEV | 开发数据库 |
| A | 分析型数据库 | TEST | 测试数据库 |
| H | 历史数据库 |  |  |

【示例】

出入系统业务生产库：AOCT、AOCT1、AOCT2

出入系统业务开发库：AOCTDEV、AOCTDEV1、AOCTDEV2

出入系统业务测试库：AOCTTEST、AOCTTEST1、AOCTTEST2

## 4.3 表命名

【规则】单表仅使用a-z、\_，分表名称为“表名\_编号

【规则】业务表名代表用途、内容：子系统简称\_业务含义\_后缀

【说明】子系统简称一般2~3字符，有开发定制简称；后缀，普通业务表不需要添加后缀。

【规则】常用后缀

|  |  |
| --- | --- |
| 类型 | 后缀 |
| 临时表 | \_tmp、\_tmp\_yyyymmdd |
| 备份表 | \_bak、\_bak\_yyyymmdd |
| 字典表 | \_dic |
| 历史归档表 | \_his |
| 日志表 | \_log |
| 类型表 | \_type |
| 大字段blob或text | \_lob |
| 代码、编码 | \_code |

【规则】常用前缀：子系统简称

* 分析型数据库

|  |  |
| --- | --- |
| 子系统 | 简称 |
| 数据存储区中的数据表 | ods\_ |
| 事实表 | fac\_ |
| 维表数据 | dim\_ |
| 中间表 | mid\_ |

## 4.4 字段命名

【规则】遵循“见名知意”的原则, 格式:名称\_后缀

【规则】布尔型的字段，以助动词(has/is)开头

【规则】常用后缀

|  |  |  |
| --- | --- | --- |
| 类型 | 后缀 | 示例 |
| 流水号/无意义主键 | id | task\_id |
| 类别 | type | station\_type |
| 名称 | name | enduser\_name |
| 描述 | desc | station\_desc |
| 代码、编码 | code | error\_code |
| 标志 | flag | debug\_flag |
| 时间 | time | insert\_time |

## 4.5 索引命名

【规则】命名格式：前缀\_表名(或缩写)\_字段名（或缩写）

【规则】复合索引，取每字段前三个字符加下划线组合

【规则】常用前缀

|  |  |
| --- | --- |
| 类型 | 前缀 |
| 主键 | pk\_ |
| UNIQUE约束 | uk\_ |
| 普通索引 | idx\_ |

## 4.6 用户

【规则】与业务名称或者数据库名称保持一致

# 5.对象设计规范

## 5.1 总则

【规则】表使用innodb存储引擎

【规则】库、表、字段必须有注释，对于冗余字段要说明其维护方法

【规则】不同表之间的相同业务意义字段的属性值要保持一致

## 5.2 库设计规范

【规则】创建DB和表时显式指定字符集

【规则】对恶魔字符集emoji用utf8mb4(ios常用)

【规则】库表字符集和前端程序、中间件必须保持一致UTF8

【禁止】禁止使用系统库存放业务数据

## 5.3 表设计规范

【规则】必须显示指定engine和charset

【规则】**所有业务表必须是innodb表**

【规则】**业务表必须有主键**

* 主键字段名有意义，字段长度要尽量小
* 主键列为独立于业务，不需要变更，业务上无意义
* 不使用组合列作为主键
* 不使用字符串列作为主键
* 不使用频繁更新的列作为主键
* 不使用UUID、MD5、HASH 作为主键
* 主键选择数值型，使用自增或全局ID，并禁止更新
* 全局ID的生成规则：前缀+流水号规则。前缀:时间戳等有序数字

【规则】热库表的数量不超过300个

【规则】热表行数纯int不超过2000W，含char不超过1000W

【规则】临时表和备份表定期清理

【规则】需要重复引用大型表或常用表中的某个数据集时，可以使用临时表

【规则】尽量保持表中行长度较小，表中的所有字段长度总和不应超过8K

【规则】表的水平切分

* 用HASH进行水平切分时，表名后缀使用十进制数，下标从000开始
* 首次分表尽量多的分，避免二次分表，二次分表的难度和成本较高
* 按日期时间分表需符合YYYYMMDDHH格式
* 采用合适的分库分表策略，如十库百表
* 大表要有明确的数据离线保留策略。

## 5.4 字段设计规范

【规则】数据类型遵循最小化原则（数据类型占位表见附录9.3）

【规则】字段的宽度适中

【禁止】禁止在数据库中使用大字段

【规则】单表字段数上限控制在20~50个

【规则】所有字段必须为NOT NULL，NULL列指定default值

【规则】单表不超过50个纯int字段，或不超过20个varchar字段

【规则】保持字段名和数据类型的一致性，例如user\_id不要使用varchar

【规则】短数据使用tinyint或smallint

【规则】使用TINYINT来代替ENUM类型

【规则】使用unsigned存储非负数值。

【规则】使用整数保存ip地址

【规则】对货币等对精度敏感的数据，应该用定点数

【规则】用DECIMAL代替FLOAT和DOUBLE存储精确浮点数

【规则】整形定义中不指定长度，比如使用INT，而不是INT(4)

【规则】变长使用varchar类型

【规则】单个字符的全部使用CHAR类型

【规则】字段中字符长度固定且较小时，使用CHAR()存储，如身份证号、手机号、性别等。

【规则】仅存储年使用YEAR类型，存储日期使用DATE类型。

【规则】精确到天，使用date类型

【规则】精确到时间（年月日时分秒），使用DATETIME 或TIMESTAMP

## 5.5 索引设计规范

【禁止】禁止外键

【规则】重要SQL需求的索引优先

【规则】单个表上索引不超过5个

【规则】重复值过高的列，不创建索引

【规则】复合索引

* 复合索引中的字段数不超过3个
* 核心SQL优先考虑索引覆盖
* 复合索引字段的顺序，区分度大的列放在前
* 合理创建复合索引，避免冗余或重复索引

【规则】不适合创建索引情况

* 更新频繁的列
* 重复值过高的列，例如性别，status, type
* 进行数学运算和函数运算的列

【规则】重要的SQL必须被索引

* SELECT、UPDATE、DELETE语句的WHERE条件列
* ORDER BY、GROUP BY、DISTINCT的字段
* 多表JOIN的关联字段

# 6.SQL开发规范

## 6.1 总则

【规则】检查语句中各字段是否存在，条件前后是否重复

【规则】尽量不要在sql中，使用1=1等开关条件

【规则】避免不必要的排序（过滤条件中排序字段已经给定值等）

【规则】避免非必要的子查询

【规则】尽量使用单条SQL操作数据库，复杂的SQL可以拆分成多个小的SQL执行

【规则】使用预编译语句（绑定变量），只传递参数

【规则】用小结果集驱动大的结果集

【规则】必须查询执行计划，严禁有大表全表扫描上线

【规则】尽可能在索引中完成排序

【规则】使用最有效的过滤条件

【规则】关键SQL语句，表关联原则上不能超过3个

【规则】SQL语句书写要规范

* SQL语句中关键字使用大写，字段名/表名等小写
* 尽量使用绑定变量，避免同一SQL由于书写格式的不同而导致多次语法解析

【规则】尽量避免向客户端返回大数据量，若数据量过大，应该考虑相应需求是否合理或使用分页分多次返回数据

【规则】程序中应有捕获SQL异常的处理机制

【规则】不要将Null与 空串''视为相同。数据库中不允许出现空串。

【规则】使用EXPLAIN判断SQL语句是否合理使用索引，避免extra列出现：Using File Sort，Using Temporary

## 6.2 DML

【规则】大批量数据dml(update、delete、insert等)操作时，需要分批次提交

【禁止】禁止单步commit，分批commit(100-10000)

【规则】insert表时建议写字段名，避免表顺序变动后字段插错顺序

【规则】尽量使用PK键值作为应用中update或delete记录条件

【规则】减少与数据库的交互次数，一次交互完成多次同类型任务

* 禁止单条SQL语句同时更新多个表
* Insert … on duplicate key update
* Replace into、insert into values(),(),()…
* Update … where id in (1,2,3,4)
* Alter table tbl\_name add column col1, add column col2

## 6.3 DDL

【禁止】在业务运行期间运行的程序中禁止ddl(create、drop等)操作

【规则】明确事务边界

* 应用中事务开始要明确关闭“自动提交”设置autocommit=0
* 应用中显式的提交或回滚事务(commit/rollback)
* 事务要简单，整个事务持续时间不能太长

## 6.4 DQL

【禁止】禁止使用select for update语句.

【禁止】结果集明确指定字段，且只列出需要字段，禁止使用select \*

【规则】From列表中，常数表优先，字典表或小表其次，大表最后

【规则】使用in代替or，in的值不超过500个

【规则】使用union all 而不是union

【规则】尽可能避免Join和子查询、or操作

【规则】多表关联所涉及字段必须有索引

## 6.5 条件

【禁止】禁止对左侧字段上使用函数、运算

【禁止】禁止出现隐式转换，明确字段与变量的类型对应

【规则】条件所涉及字段上尽量有索引

【规则】字段顺序尽量与索引顺序保持一致

【规则】避免使用 not、!=、<>、is null, is not null条件

【规则】多用等值操作；>,<操作尽量转换为>=,<=

【规则】避免使用NULL字段做为条件，不能使用索引

在字段上设置默认值（无意义值），确保表中列没有null值

【规则】like子句尽量前置匹配，前置不加%，以便能够利用索引，必要时可以考虑将列拆分或列值倒序后插入

【规则】充分利用最左前缀索引

【禁止】禁止无意义order by操作(where条件已给定值等)

【禁止】禁止使用order by rand()

【规则】尽可能减少group by操作

【规则】group by 语句如果得到的结果集，需要sort ,加上order by

【规则】尽量使用where条件替代having子句

【规则】使用LIMIT分页时，偏移量不能超过500

# 7.行为规范

## 7.1 建表申请

* 必须提供建表用途及建表申请人，描述清楚所属业务模块及表中数据用途
* 必须提供表中数据的预估量，以及每行数据大致占用空间
* 必须严格区分表的用途，并在提交脚本中说明，诸如：日志类表、业务信息表、配置表、临时表等
* 必须清楚标明每张表中数据的生命周期，对于日志类和接口类表数据生命周期不得超过6个月
* 必须在设计伊始就考虑数据清理机制
* 超过数据生命周期的表中数据将会移出生产环境，在程序设计中必须考虑这个问题
* 提交建表、修改表结构需求时，必须详细注明所有相关SQL语句
* 所有的建表、修改表结构需要确定建立哪些索引后才可以上线

## 7.2 重大操作

* 所有的修改表结构、加索引操作都需要将涉及到所改表的SQL发出来，告知DBA等相关人员
* 在建新表加字段之前，要求研发至少要提前至少3天邮件出来，给出dba评估、优化和审核的时间
* 批量导入、导出数据必须提前通知DBA协助观察
* 批量更新数据，如update, delete 操作，需要DBA进行审查，并在执行过程中观察服务

## 7.3 危险行为

* 禁止在主库上执行后台管理和统计类的功能查询
* 禁止有super权限的业务账号存在
* 业务出现故障时，及时通知DBA协助排查
* 数据库数据丢失，及时联系DBA进行恢复
* 对单表的多次alter操作须合并为一次操作
* 不在MySQL数据库中存放业务逻辑
* 重要的库表，提前与DBA沟通确定维护和备份优先级
* 不在业务高峰期批量更新、查询数据库
* 不在业务高峰期修改表结构，添加、删除字段

# 8.数据库脚本

## 8.1 建表脚本

数据库的建表脚本作为数据库设计的规范产出物，需要严格进行版本控制，有统一的命名规范，命名规范如下：

* 创建表脚本：项目简称\_create\_table.sql。
* 修改表脚本：项目简称\_alter\_table.sql，在原有表的基础上做的修改，增加字段、主键、外键、索引等。

注：修改表脚本只作为备忘，所有表结构的修改，都必须实时更新“创建表脚本”，便于项目实施时表结构的导入。

# 9.附录

## 9.1 MySQL保留关键字

对象命名不允许使用的关键字：

|  |  |  |  |
| --- | --- | --- | --- |
| ABSOLUTE | INDICATOR | SYSTEM\_USER | AUXILIARY |
| ABSOLUTE | INITIALIZE | TABLE | BUFFERPOOL |
| ACTION | INITIALLY | TEMPORARY | CALLED |
| ADA | INNER | TERMINATE | CAPTURE |
| ADD | INOUT | TEXTSIZE | CCSID |
| ADMIN | INPUT | THAN | CLUSTER |
| AFTER | INSENSITIVE | THEN | COLLECTION |
| AGGREGATE | INSERT | TIME | COLLID |
| ALIAS | INT | TIMESTAMP | COMMENT |
| ALL | INTEGER | TIMEZONE\_HOUR | CONCAT |
| ALLOCATE | INTERSECT | TIMEZONE\_MINUTE | CONDITION |
| ALTER | INTERVAL | TO | COUNT\_BIG |
| AND | INTO | TOP | CURRENT\_LC\_PATH |
| ANY | IS | TRAILING | CURRENT\_SERVER |
| ARE | ISOLATION | TRAN | CURRENT\_TIMEZONE |
| ARRAY | ITERATE | TRANSACTION | DAYS |
| AS | JOIN | TRANSLATE | DB2GENERAL |
| ASC | KEY | TRANSLATION | DB2SQL |
| ASSERTION | KILL | TREAT | DBA |
| AT | LANGUAGE | TRIGGER | DBINFO |
| AUTHORIZATION | LARGE | TRIM | DBSPACE |
| AVG | LAST | TRUNCATE | DISALLOW |
| BACKUP | LATERAL | TSEQUAL | DO |
| BEFORE | LEADING | UNDER | DSSIZE |
| BEGIN | LEFT | UNION | EDITPROC |
| BETWEEN | LESS | UNIQUE | ELSEIF |
| BINARY | LEVEL | UNKNOWN | ENCRYPT |
| BIT | LIKE | UNNEST | ERASE |
| BIT\_LENGTH | LIMIT | UPDATE | EXCLUSIVE |
| BLOB | LINENO | UPDATETEXT | EXPLAIN |
| BOOLEAN | LOAD | UPPER | FENCED |
| BOTH | LOCAL | USAGE | FIELDPROC |
| BREADTH | LOCALTIME | USE | FINAL |
| BREAK | LOCALTIMESTAMP | USER | GENERATED |
| BROWSE | LOCATOR | USING | GRAPHIC |
| BULK | LOWER | VALUE | HANDLER |
| BY | MAP | VALUES | HOURS |
| CALL | MATCH | VARCHAR | IDENTIFIED |
| CASCADE | MAX | VARIABLE | INTEGRITY |
| CASCADED | MIN | VARYING | ISOBID |
| CASE | MINUTE | VIEW | JAVA |
| CAST | MODIFIES | WAITFOR | LABEL |
| CATALOG | MODIFY | WHEN | LC\_CTYPE |
| CHAR | MODULE | WHENEVER | LEAVE |
| CHAR\_LENGTH | MONTH | WHERE | LINKTYPE |
| CHARACTER | NAMES | WHILE | LOCALE |
| CHARACTER\_LENGTH | NATIONAL | WITH | LOCATORS |
| CHECK | NATURAL | WITHOUT | LOCK |
| CHECKPOINT | NCHAR | WORK | LOCKSIZE |
| CLASS | NCLOB | WRITE | LONG |
| CLOB | NEW | WRITETEXT | LOOP |
| CLOSE | NEXT | YEAR | MICROSECOND |
| CLUSTERED | NO | ZONE | MICROSECONDS |
| COALESCE | NOCHECK | FALSE | MINUTES |
| COLLATE | NONCLUSTERED | TRUE | MODE |
| COLLATION | NONE | ACCESS | MONTHS |
| COLUMN | NOT | ADD | NAME |
| COMMIT | NULL | ALL | NAMED |
| COMPLETION | NULLIF | ALTER | NHEADER |
| COMPUTE | NUMERIC | AND | NODENAME |
| CONNECT | OBJECT | ANY | NODENUMBER |
| CONNECTION | OCTET\_LENGTH | AS | NULLS |
| CONSTRAINT | OF | ASC | NUMPARTS |
| CONSTRAINTS | OFF | AUDIT | OBID |
| CONSTRUCTOR | OFFSETS | BETWEEN | OPTIMIZATION |
| CONTAINS | OLD | BY | OPTIMIZE |
| CONTAINSTABLE | ON | CHAR | PACKAGE |
| CONTINUE | ONLY | CHECK | PAGE |
| CONVERT | OPEN | CLUSTER | PAGES |
| CORRESPONDING | OPENDATASOURCE | COLUMN | PART |
| COUNT | OPENQUERY | COMMENT | PARTITION |
| CREATE | OPENROWSET | COMPRESS | PCTFREE |
| CROSS | OPENXML | CONNECT | PCTINDEX |
| CUBE | OPERATION | CREATE | PIECESIZE |
| CURRENT | OPTION | CURRENT | PRIQTY |
| CURRENT\_DATE | OR | DATE | PRIVATE |
| CURRENT\_PATH | ORDER | DECIMAL | PROGRAM |
| CURRENT\_ROLE | ORDINALITY | DEFAULT | PSID |
| CURRENT\_TIME | OUT | DELETE | QUERYNO |
| CURRENT\_TIMESTAMP | OUTER | DESC | RECOVERY |
| CURRENT\_USER | OUTPUT | DISTINCT | RELEASE |
| CURSOR | OVER | DROP | RENAME |
| CYCLE | OVERLAPS | ELSE | REORG |
| DATA | PAD | EXCLUSIVE | REPEAT |
| DATABASE | PARAMETER | EXISTS | RESET |
| DATE | PARAMETERS | FILE | RESOURCE |
| DAY | PARTIAL | FLOAT | RRN |
| DBCC | PASCAL | FOR | RUN |
| DEALLOCATE | PATH | FROM | SCHEDULE |
| DEC | PERCENT | GRANT | SCRATCHPAD |
| DECIMAL | PLAN | GROUP | SECONDS |
| DECLARE | POSITION | HAVING | SECQTY |
| DEFAULT | POSTFIX | IDENTIFIED | SECURITY |
| DEFERRABLE | PRECISION | IMMEDIATE | SHARE |
| DEFERRED | PREFIX | IN | SIMPLE |
| DELETE | PREORDER | INCREMENT | SOURCE |
| DENY | PREPARE | INDEX | STANDARD |
| DEPTH | PRESERVE | INITIAL | STAY |
| DEREF | PRIMARY | INSERT | STOGROUP |
| DESC | PRINT | INTEGER | STORES |
| DESCRIBE | PRIOR | INTERSECT | STORPOOL |
| DESCRIPTOR | PRIVILEGES | INTO | STYLE |
| DESTROY | PROC | IS | SUBPAGES |
| DESTRUCTOR | PROCEDURE | LEVEL | SYNONYM |
| DETERMINISTIC | PUBLIC | LIKE | TABLESPACE |
| DIAGNOSTICS | RAISERROR | LOCK | TYPE |
| DICTIONARY | READ | LONG | UNDO |
| DISCONNECT | READS | MAXEXTENTS | UNTIL |
| DISK | READTEXT | MINUS | UPSERT |
| DISTINCT | REAL | MLSLABEL | VALIDPROC |
| DISTRIBUTED | RECONFIGURE | MODE | VARIANT |
| DOMAIN | RECURSIVE | MODIFY | VCAT |
| DOUBLE | REF | NOAUDIT | VOLUMES |
| DROP | REFERENCES | NOCOMPRESS | WLM |
| DUMMY | REFERENCING | NOT | YEARS |
| DUMP | RELATIVE | NOWAIT | ADD |
| DYNAMIC | REPLICATION | NULL | ARITH\_OVERFLOW |
| EACH | RESTORE | NUMBER | CHAR\_CONVERT |
| ELSE | RESTRICT | OF | CONFIRM |
| END | RESULT | OFFLINE | CONTROLROW |
| END-EXEC | RETURN | ON | ENDTRAN |
| EQUALS | RETURNS | ONLINE | ERRORDATA |
| ERRLVL | REVOKE | OPTION | ERROREXIT |
| ESCAPE | RIGHT | OR | EXP\_ROW\_SIZE |
| EVERY | ROLE | ORDER | IDENTITY\_GAP |
| EXCEPT | ROLLBACK | PCTFREE | IDENTITY\_START |
| EXCEPTION | ROLLUP | PRIOR | INSTALL |
| EXEC | ROUTINE | PRIVILEGES | JAR |
| EXECUTE | ROW | PUBLIC | MAX\_ROWS\_PER\_PAGE |
| EXISTS | ROWCOUNT | RAW | MIRROR |
| EXIT | ROWGUIDCOL | RENAME | MIRROREXIT |
| EXTERNAL | ROWS | RESOURCE | NOHOLDLOCK |
| EXTRACT | RULE | REVOKE | NUMERIC\_TRUNCATION |
| FETCH | SAVE | ROW | ONCE |
| FILE | SAVEPOINT | ROWID | ONLINE |
| FILLFACTOR | SCHEMA | ROWNUM | PERM |
| FIRST | SCOPE | ROWS | PERMANENT |
| FLASE | SCROLL | SELECT | PROCESSEXIT |
| FLOAT | SEARCH | SESSION | PROXY\_TABLE |
| FOR | SECOND | SET | QUIESCE |
| FOREIGN | SECTION | SHARE | READPAST |
| FORTRAN | SELECT | SIZE | REMOVE |
| FOUND | SEQUENCE | SMALLINT | REPLACE |
| FREE | SESSION | START | RESERVEPAGEGAP |
| FREETEXT | SESSION\_USER | SUCCESSFUL | SHARED |
| FREETEXTTABLE | SET | SYNONYM | STRIPE |
| FROM | SETS | SYSDATE | SYB\_IDENTITY |
| FULL | SETUSER | TABLE | SYB\_RESTREE |
| FUNCTION | SHUTDOWN | THEN | TEMP |
| GENERAL | SIZE | TO | UNPARTITION |
| GET | SMALLINT | TRIGGER | USER\_OPTION |
| GLOBAL | SOME | UID |  |
| GO | SPACE | UNION |  |
| GOTO | SPECIFIC | UNIQUE |  |
| GRANT | SPECIFICTYPE | UPDATE |  |
| GROUP | SQL | USER |  |
| GROUPING | SQLCA | VALIDATE |  |
| HAVING | SQLCODE | VALUES |  |
| HOLDLOCK | SQLERROR | VARCHAR |  |
| HOST | SQLEXCEPTION | VARCHAR2 |  |
| HOUR | SQLSTATE | VIEW |  |
| IDENTITY | SQLWARNING | WHENEVER |  |
| IDENTITY\_INSERT | START | WHERE |  |
| IDENTITYCOL | STATE | WITH |  |
| IF | STATEMENT | ACQUIRE |  |
| IGNORE | STATIC | ALLOW |  |
| IMMEDIATE | STATISTICS | AS |  |
| IN | STRUCTURE | ASUTIME |  |
| INCLUDE | SUBSTRING | AUDIT |  |
| INDEX | SUM | AUX |  |

## 9.2 字段类型策略

常用字段类型

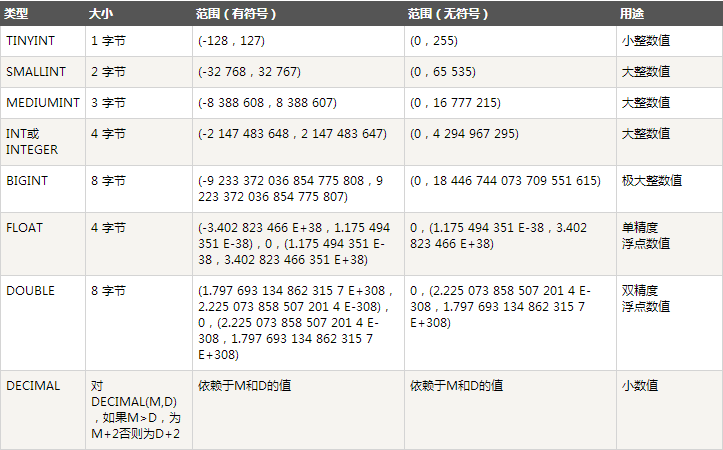
|  |  |
| --- | --- |
| 常用字段 | 类型 |
| 销售额等金额 | decimal (16,2) |
| 税率、比例、分成 | decimal (10,6) |
| 货物单价 | decimal (16,6) |
| 人数 | INT |
| 人名 | VARCHAR(50) |
| 单位名称、地址 | VARCHAR (100) |
| 说明、理由、意见 | VARCHAR (200) |

常用字段名

|  |  |
| --- | --- |
| **常用字段名** | **解释** |
| optr\_code | 操作员工号 |
| opt\_date | 操作时间 |
| remark | 备用字段 |
| stand | 备注 |
| is\_ | 是、否”类型的字段 |

## 9.3 数据类型占位对照表

数值型



字符型



日期/时间类型

