Mysql数据与交互设计

数据库的分库分为水平分库和垂直分库两种，当前使用的垂直分库如下示例：

1.数据库user\_manager：

/\*用户列表\*/  
SET FOREIGN\_KEY\_CHECKS=0;  
DROP TABLE IF EXISTS `data\_user`;  
CREATE TABLE `data\_user` (

`user\_id` bigint(32) NOT NULL COMMENT '32用户主键',

`nick\_name` varchar(32) NOT NULL COMMENT '32用户昵称',

`user\_name` char(12) NOT NULL COMMENT '12登录用户名',

`password` char(16) NOT NULL COMMENT '16登录密码',

`role\_name` varchar(32) DEFAULT NULL COMMENT '32角色名称',

`role\_code` varchar(32) DEFAULT NULL COMMENT '32角色编码',

`certify\_identity` tinyint(1) NOT NULL COMMENT '1是否通过身份证明',

`mobile\_phone` bigint(11) DEFAULT NULL COMMENT '11手机',

`telephone` bigint(8) DEFAULT NULL COMMENT '8电话',

`job` varchar(64) DEFAULT NULL COMMENT '64职务',

`job\_unit` varchar(64) DEFAULT NULL COMMENT '64工作单位',

PRIMARY KEY (`user\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='用户列表';

2.数据库ware\_manager：

/\*商品家族表\*/  
DROP TABLE IF EXISTS `data\_ware\_family`;  
CREATE TABLE `data\_ware\_family` (

`ware\_family\_id` bigint(32) NOT NULL COMMENT '商品家族主键',

`ware\_family\_name` varchar(16) NOT NULL COMMENT '家族名称',

`ware\_family\_describe` varchar(32) DEFAULT NULL COMMENT '家族描述',

PRIMARY KEY (`ware\_family\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='商品家族列表';

/\*商品类型表\*/  
DROP TABLE IF EXISTS `data\_ware\_type`;  
CREATE TABLE `data\_ware\_type` (

`ware\_type\_id` bigint(32) NOT NULL COMMENT '商品类型主键',

`ware\_type\_name` varchar(16) NOT NULL COMMENT '类型名称',

`ware\_family\_id` bigint(32) NOT NULL COMMENT '商品家族主键冗余',

`ware\_type\_describe` varchar(64) DEFAULT NULL COMMENT '类型描述',

PRIMARY KEY (`ware\_type\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='商品类型表';

/\*单一商品表\*/  
DROP TABLE IF EXISTS `data\_single\_ware`;  
CREATE TABLE `data\_single\_ware` (

`single\_ware\_id` bigint(32) NOT NULL COMMENT '商品主键',

`ware\_type\_id` bigint(32) NOT NULL COMMENT '商品类型主键冗余',

`single\_ware\_describe` varchar(128) DEFAULT NULL COMMENT '商品详细描述',

`single\_ware\_name` varchar(64) NOT NULL COMMENT '商品名称',

`single\_ware\_code` varchar(32) DEFAULT NULL COMMENT '商品编码',

`single\_ware\_price` double(12,0) NOT NULL COMMENT '商品单价（人民币：元）',

`single\_ware\_long` double(8,0) DEFAULT NULL COMMENT '长（米）',

`single\_ware\_width` double(8,0) DEFAULT NULL COMMENT '宽（米）',

`single\_ware\_height` double(8,0) DEFAULT NULL COMMENT '高（米）',

`single\_ware\_acreage` double(8,0) DEFAULT NULL COMMENT '面积（平方米）',

`single\_ware\_volunm` double(8,0) DEFAULT NULL COMMENT '体积（立方米）',

`single\_ware\_weight` double(8,0) DEFAULT NULL COMMENT '重量（千克）',

PRIMARY KEY (`single\_ware\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='单一商品表';

3.数据库shop\_manager：

/\*商店列表\*/  
SET FOREIGN\_KEY\_CHECKS=0;  
DROP TABLE IF EXISTS `data\_shop`;  
CREATE TABLE `data\_shop` (

`shop\_id` bigint(32) NOT NULL COMMENT '32商店主键',

`shop\_name` varchar(32) NOT NULL COMMENT '32商店名称',

`shop\_type` varchar(16) NOT NULL COMMENT '16商店类型（旗舰店[在售品牌唯一:需要品牌认证]、专营店[在售品类唯一]、厂家直销[需要厂家认证]、个体户...）',

`shop\_business\_id` bigint(32) NOT NULL COMMENT '32商点归属用户主键冗余',

`shop\_business\_name` varchar(32) NOT NULL COMMENT '32商店归属用户user名称冗余',

`shop\_describe` varchar(128) DEFAULT NULL COMMENT '128商店描述',

PRIMARY KEY (`shop\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='商店列表';

/\*店内在售品类表\*/  
SET FOREIGN\_KEY\_CHECKS=0;  
DROP TABLE IF EXISTS `data\_sale\_type`;  
CREATE TABLE `data\_sale\_type` (

`sale\_type\_id` bigint(32) NOT NULL COMMENT '32在售品类主键',

`shop\_id` bigint(32) NOT NULL COMMENT '32商家主键冗余',

`ware\_family\_id` bigint(32) NOT NULL COMMENT '32商品家族主键冗余',

`ware\_family\_name` varchar(16) NOT NULL COMMENT '16商品家族名称冗余',

`ware\_type\_id` bigint(32) NOT NULL COMMENT '32商品类型主键冗余',

`ware\_type\_name` varchar(16) NOT NULL COMMENT '16商品类型名称冗余',

PRIMARY KEY (`sale\_type\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='店内在售品类表';

/\*售类内在售单品表\*/  
DROP TABLE IF EXISTS `data\_single\_sale`;  
CREATE TABLE `data\_single\_sale` (

`single\_sale\_id` bigint(32) NOT NULL COMMENT '32在售单品主键',

`sale\_type\_id` bigint(32) NOT NULL COMMENT '32在售品类主键冗余',

`single\_ware\_id` bigint(32) NOT NULL COMMENT '32单一商品主键冗余',

`single\_ware\_name` varchar(64) NOT NULL COMMENT '64单一商品名称冗余',

PRIMARY KEY (`single\_sale\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='售类内在售单品表';

4.数据库order\_manager：

/\*订单列表\*/  
SET FOREIGN\_KEY\_CHECKS=0;  
DROP TABLE IF EXISTS `data\_order`;  
CREATE TABLE `data\_order` (

`order\_id` bigint(32) NOT NULL COMMENT '订单号',

`order\_customer\_id` bigint(32) NOT NULL COMMENT '订单归属人user主键冗余',

`order\_customer\_name` varchar(32) NOT NULL COMMENT '订单归属人user名称冗余',

`order\_business\_id` bigint(32) NOT NULL COMMENT '订单来源商家user主键冗余',

`order\_business\_name` varchar(32) NOT NULL COMMENT '订单来源商家user名称冗余',

`order\_origin\_shop\_id` bigint(32) NOT NULL COMMENT '订单来源商店主键冗余',

`order\_origin\_shop\_name` varchar(32) NOT NULL COMMENT '订单来源商店名称冗余',

`order\_amount` double(14,0) NOT NULL COMMENT '订单总金额',

`receiver\_name` varchar(32) NOT NULL COMMENT '32收货人名称',

`receiver\_sex` varchar(4) NOT NULL COMMENT '4收货人性别',

`receiver\_mobile` bigint(11) NOT NULL COMMENT '11收货人手机',

`receiver\_tel` bigint(8) DEFAULT NULL COMMENT '8收货人电话',

`receiver\_province` varchar(16) NOT NULL COMMENT '16省份',

`receiver\_city` varchar(32) NOT NULL COMMENT '32市',

`receiver\_district` varchar(32) NOT NULL COMMENT '32区县',

`receiver\_street` varchar(64) NOT NULL COMMENT '64街道门牌号',

`receiver\_postcode` bigint(8) NOT NULL COMMENT '8邮编',

PRIMARY KEY (`order\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='订单列表';

/\*费用列表\*/  
SET FOREIGN\_KEY\_CHECKS=0;  
DROP TABLE IF EXISTS `data\_expense`;  
CREATE TABLE `data\_expense` (

`expense\_id` bigint(32) NOT NULL COMMENT '费用主键',

`order\_id` bigint(32) NOT NULL COMMENT '订单主键冗余',

`expense\_amount` double(12,0) NOT NULL COMMENT '费用总金额',

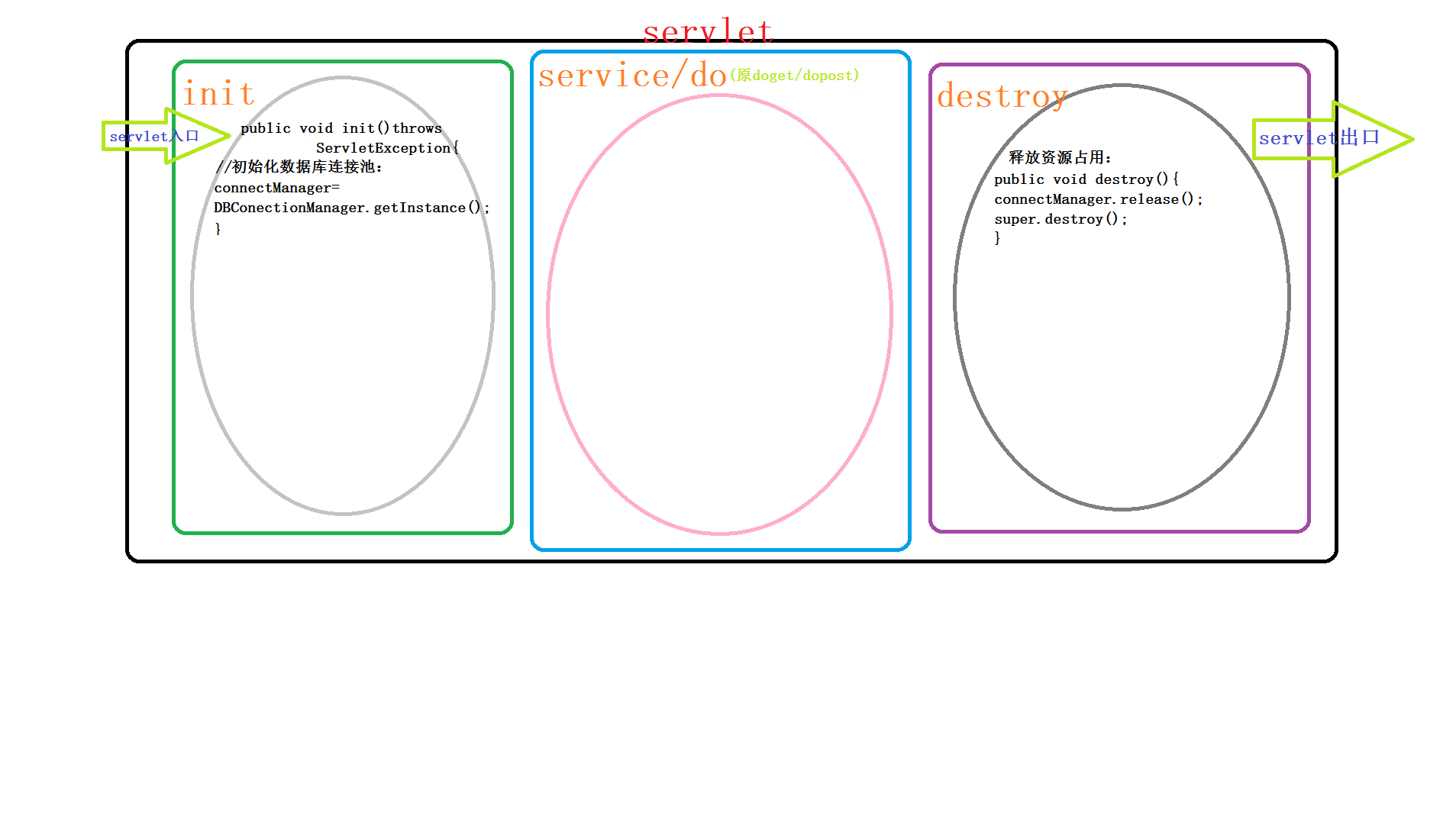
`single\_ware\_id` bigint(32) NOT NULL COMMENT '商品单项主键冗余',

`single\_ware\_quality` double(14,0) NOT NULL COMMENT '商品单项数量',

PRIMARY KEY (`expense\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='费用列表';

数据库连接池与servlet的结合使用

一般的数据库分库最好采用垂直分库，而分库的大小、库内内容等分配策略需要在预测将来的业务需求可能规模而制定；相对而言水平分库会显得较为复杂且更不好管理；例如：把order\_manager分为order\_manager1、order\_manager2**... ...**

为了数据库进行更新的时候避免冲突，采用版本号version作为乐观锁标志来限制用户对每张表的更新操作。

对于用户用到的注册、创建等操作需要唯一主键id时，使用雪花算法的分布式唯一主键生成策略，生成的id是32位的递增整型数据。根据实际情况，类似于订单表、费用单项表这样的可以考虑直接使用数据库自增。

当单张表的数据量不足以支撑业务数据量时当然是要使用数据库分表策略啦，既然每个用户的id是由32位的int组成的，那我们在执行表操作前就可以通过对32的int进行取余%操作（取余数根据需要分的表数量，被取余数就是用户id），得到的余数就是要进行操作的第几张表，比如：data\_order1、data\_order2**... ...**

## Mysql版本更新后启动遇到的错误

Can't connect to MySQL server on '127.0.0.1' (10061)

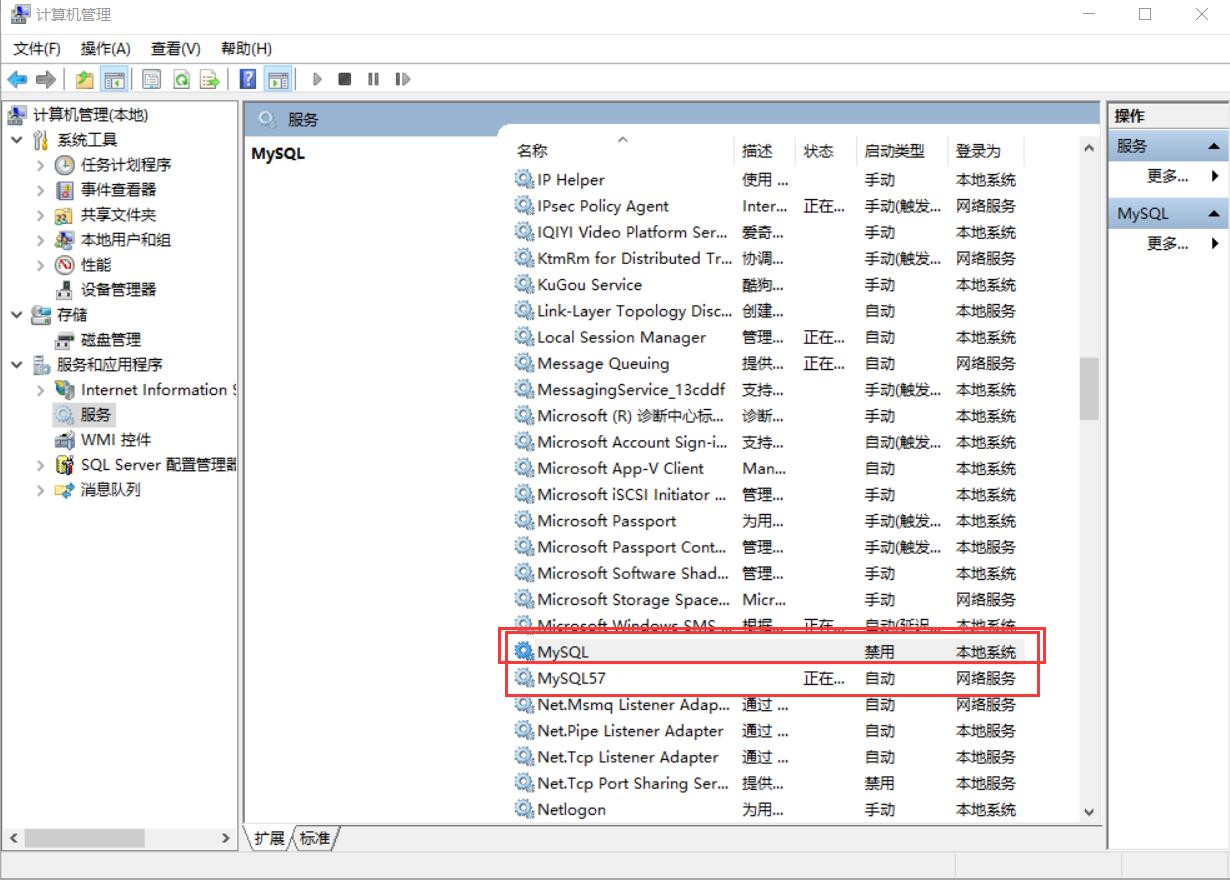
参考：

<http://www.jb51.net/article/26505.htm>

<https://www.cnblogs.com/technologylife/p/6010205.html>

<https://www.cnblogs.com/jinjiangongzuoshi/p/3696904.html>

解决：



当旧版本mysql卸载后，服务并未被移除，在两个版本mysql都为自启动时，按顺序第一个服务找不到启动文件报错。

## 参考：

### 大型互联网应用面对海量访问时，数据库往往是最容易成为性能瓶颈的地方，故而数据库的设计就显得尤为重要。除了表和字段结构的设计上需要再三考虑谨慎、应用访问数据库的执行语句需要考虑次数和性能效率之外，往往都需要进行分库分表的策略分析和设计。

分库分表伴随的完整性和一致性问题<https://www.cnblogs.com/aigongsi/archive/2013/01/25/2875731.html>

**分表与分库设计方式** <http://blog.csdn.net/winy_lm/article/details/50708493>

### 为了数据库进行更新的时候避免冲突，采用版本号version作为乐观锁标志来限制用户的更新操作。

[乐观锁的使用](http://www.cnblogs.com/DengGao/p/6479824.html) <https://www.cnblogs.com/DengGao/p/6479824.html>

### 对于用户用到的注册、创建等操作需要唯一主键id时，根据实际考虑使用雪花算法的分布式唯一主键生成策略

Java实现SnowFlake <http://blog.csdn.net/h254532699/article/details/54312915>

[Twitter的snowflake (Java版)](http://www.cnblogs.com/relucent/p/4955340.html)<https://www.cnblogs.com/relucent/p/4955340.html>

Java 雪花算法<https://jingyan.baidu.com/article/335530da86351919cb41c32a.html>