

# 尚硅谷大数据项目之尚品汇（业务数据采集平台）

（作者：尚硅谷研究院）

版本：V4.1.0

## 第 1 章 电商业务简介

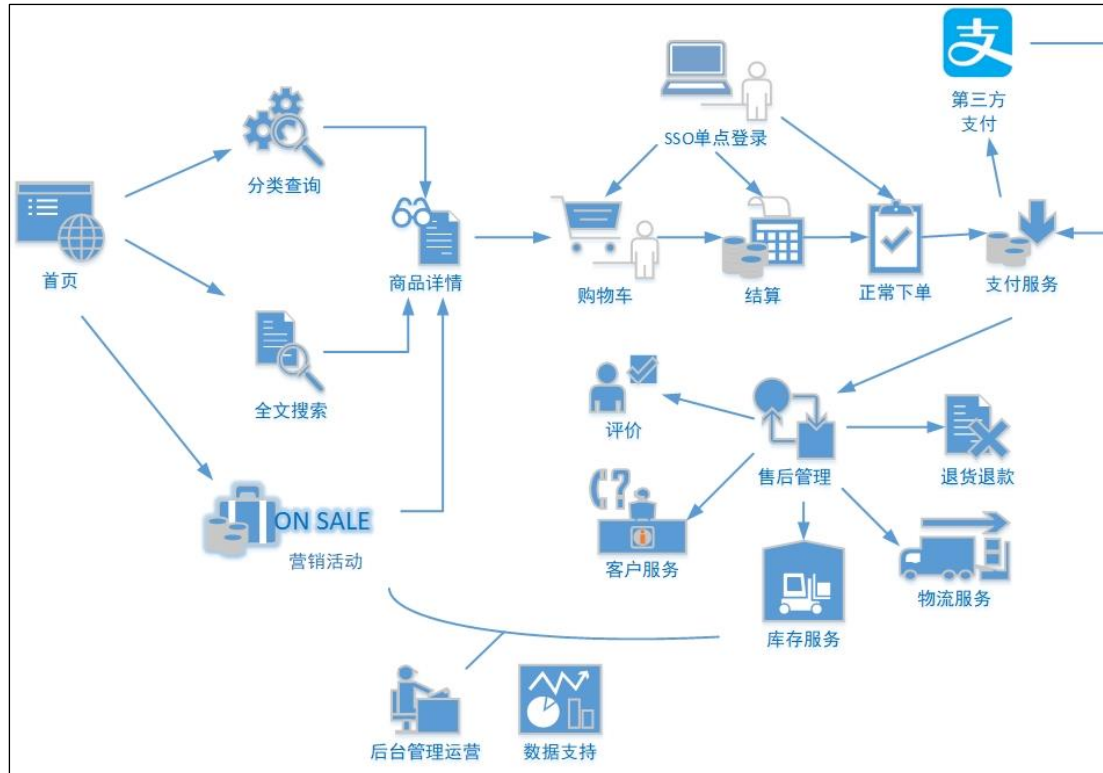
### 1.1 电商业务流程

电商的业务流程可以以一个普通用户的浏览足迹为例进行说明，用户点开电商首页开始浏览，可能会通过分类查询也可能通过全文搜索寻找自己中意的商品，这些商品无疑都是存储在后台的管理系统中的。

当用户寻找到自己中意的商品，可能会想要购买，将商品添加到购物车后发现需要登录，登录后对商品进行结算，这时候购物车的管理和商品订单信息的生成都会对业务数据库产生影响，会生成相应的订单数据和支付数据。

订单正式生成之后，还会对订单进行跟踪处理，直到订单全部完成。

电商的主要业务流程包括用户前台浏览商品时的商品详情的管理，用户商品加入购物车进行支付时用户个人中心&支付服务的管理，用户支付完成后订单后台服务的管理，这些流程涉及到了十几个甚至几十个业务数据表，甚至更多。



## 1.2 电商常识

### 1.2.1 SKU 和 SPU

**SKU = Stock Keeping Unit (库存量基本单位)**。现在已经被引申为产品统一编号的简称，每种产品均对应应有唯一的 SKU 号。

**SPU (Standard Product Unit)**：是商品信息聚合的最小单位，是一组可复用、易检索的标准化信息集合。

例如：**iPhoneX 手机就是 SPU**。一台银色、128G 内存的、支持联通网络的 iPhoneX，就是 SKU。



SPU 表示一类商品。同一 SPU 的商品可以共用商品图片、海报、销售属性等。

更多 [Java - 大数据 - 前端 - python 人工智能资料下载](#)，可百度访问：[尚硅谷官网](#)

## 1.2.2 平台属性和销售属性

### 1) 平台属性

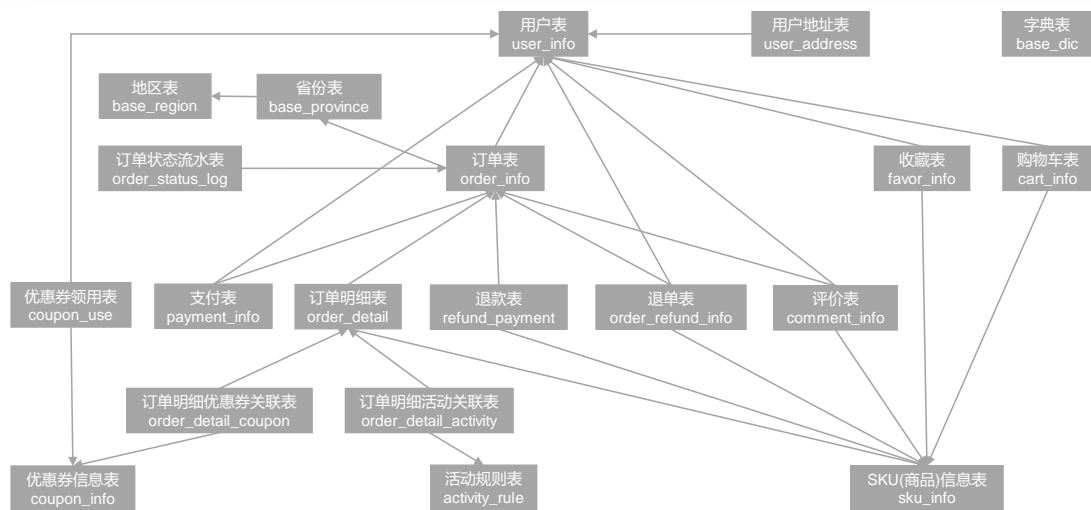


### 2) 销售属性

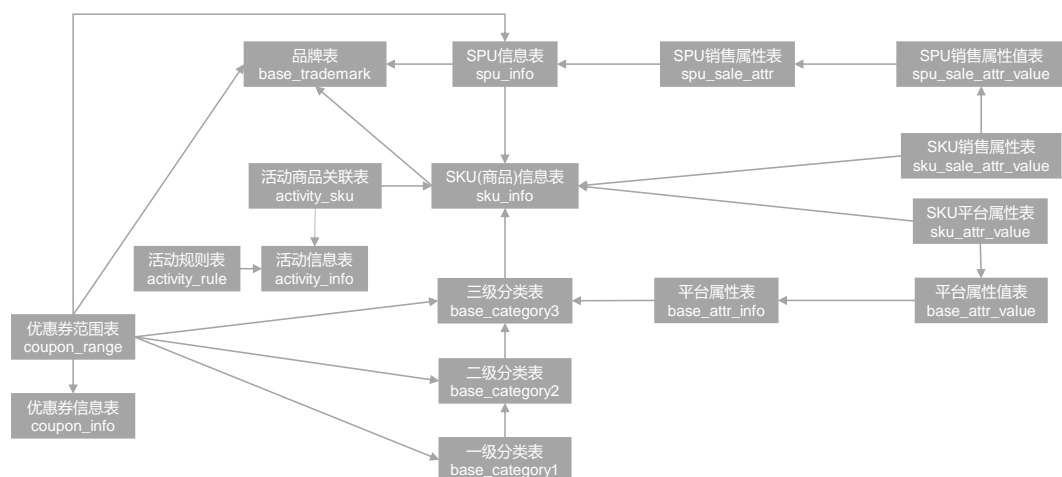


## 1.3 电商系统表结构

以下为本电商数仓系统涉及到的业务数据表结构关系。这 34 个表以订单表、用户表、SKU 商品表、活动表和优惠券表为中心，延伸出了优惠券领用表、支付流水表、活动订单表、订单详情表、订单状态表、商品评论表、编码字典表退单表、SPU 商品表等，用户表提供用户的详细信息，支付流水表提供该订单的支付详情，订单详情表提供订单的商品数量等情况，商品表给订单详情表提供商品的详细信息。本次讲解以此 34 个表为例，实际项目中，业务数据库中表格远远不止这些。



让天下没有难学的技术



让天下没有难学的技术

### 1.3.1 活动信息表 (activity\_info)

字段名	字段说明
id	活动 id
activity_name	活动名称
activity_type	活动类型（1：满减，2：折扣）
activity_desc	活动描述
start_time	开始时间
end_time	结束时间
create_time	创建时间

### 1.3.2 活动规则表 (activity\_rule)

id	编号
activity_id	活动 ID
activity_type	活动类型
condition_amount	满减金额
condition_num	满减件数
benefit_amount	优惠金额
benefit_discount	优惠折扣
benefit_level	优惠级别

### 1.3.3 活动商品关联表 (activity\_sku)

字段名	字段说明
id	编号
activity_id	活动 id
sku_id	sku_id
create_time	创建时间

### 1.3.4 平台属性表 (base\_attr\_info)

字段名	字段说明
id	编号
attr_name	属性名称
category_id	分类 id
category_level	分类层级

### 1.3.5 平台属性值表 (base\_attr\_value)

字段名	字段说明
id	编号
value_name	属性值名称
attr_id	属性 id

### 1.3.6 一级分类表 (base\_category1)

字段名	字段说明
id	编号
name	分类名称

### 1.3.7 二级分类表 (base\_category2)

字段名	字段说明
id	编号
name	二级分类名称
category1_id	一级分类编号

### 1.3.8 三级分类表 (base\_category3)

字段名	字段说明
id	编号
name	三级分类名称
category2_id	二级分类编号

### 1.3.9 字典表 (base\_dic)

字段名	字段说明
dic_code	编号
dic_name	编码名称
parent_code	父编号
create_time	创建日期
operate_time	修改日期

### 1.3.10 省份表 (base\_province)

字段名	字段说明
id	id
name	省名称
region_id	大区 id
area_code	行政区位码
iso_code	国际编码
iso_3166_2	ISO3166 编码

### 1.3.11 地区表 (base\_region)

字段名	字段说明
id	大区 id
region_name	大区名称

### 1.3.12 品牌表 (base\_trademark)

字段名	字段说明
id	编号
tm_name	属性值
logo_url	品牌 logo 的图片路径

### 1.3.13 购物车表 (cart\_info)

字段名	字段说明
id	编号
user_id	用户 id
sku_id	skuid
cart_price	放入购物车时价格
sku_num	数量
img_url	图片文件

sku_name	sku 名称 (冗余)
is_checked	是否已经下单
create_time	创建时间
operate_time	修改时间
is_ordered	是否已经下单
order_time	下单时间
source_type	来源类型
source_id	来源编号

### 1.3.14 评价表 (comment\_info)

字段名	字段说明
id	编号
user_id	用户 id
nick_name	用户昵称
head_img	图片
sku_id	商品 sku_id
spu_id	商品 spu_id
order_id	订单编号
appraise	评价 1 好评 2 中评 3 差评
comment_txt	评价内容
create_time	创建时间
operate_time	修改时间

### 1.3.15 优惠券信息表 (coupon\_info)

字段名	字段说明
id	购物券编号
coupon_name	购物券名称
coupon_type	购物券类型 1 现金券 2 折扣券 3 满减券 4 满件打折券
condition_amount	满额数 (3)
condition_num	满件数 (4)
activity_id	活动编号
benefit_amount	减金额 (13)
benefit_discount	折扣 (24)
create_time	创建时间
range_type	范围类型 1、商品(spuid) 2、品类(三级分类 id) 3、品牌
limit_num	最多领用次数
taken_count	已领用次数
start_time	可以领取的开始日期
end_time	可以领取的结束日期
operate_time	修改时间
expire_time	过期时间
range_desc	范围描述

### 1.3.16 优惠券优惠范围表 (coupon\_range)

字段名	字段说明
id	购物券编号
coupon_id	优惠券 id
range_type	范围类型 1、商品(spuid) 2、品类(三级分类 id) 3、品牌
range_id	范围 id

### 1.3.17 优惠券领用表 (coupon\_use)

字段名	字段说明
id	编号
coupon_id	购物券 ID
user_id	用户 ID
order_id	订单 ID
coupon_status	购物券状态 (1: 未使用 2: 已使用)
get_time	获取时间
using_time	使用时间
used_time	支付时间
expire_time	过期时间

### 1.3.18 收藏表 (favor\_info)

字段名	字段说明
id	编号
user_id	用户 id
sku_id	skuid
spu_id	商品 id
is_cancel	是否已取消 0 正常 1 已取消
create_time	创建时间
cancel_time	修改时间

### 1.3.19 订单明细表 (order\_detail)

字段名	字段说明
id	编号
order_id	订单编号
sku_id	sku_id
sku_name	sku 名称 (冗余)
img_url	图片名称 (冗余)
order_price	购买价格(下单时 sku 价格)
sku_num	购买个数
create_time	创建时间
source_type	来源类型
source_id	来源编号
split_total_amount	分摊总金额



split_activity_amount	分摊活动减免金额
split_coupon_amount	分摊优惠券减免金额

### 1.3.20 订单明细活动关联表 (order\_detail\_activity)

字段名	字段说明
id	编号
order_id	订单 id
order_detail_id	订单明细 id
activity_id	活动 ID
activity_rule_id	活动规则
sku_id	skuID
create_time	获取时间

### 1.3.21 订单明细优惠券关联表 (order\_detail\_coupon)

字段名	字段说明
id	编号
order_id	订单 id
order_detail_id	订单明细 id
coupon_id	购物券 ID
coupon_use_id	购物券领用 id
sku_id	skuID
create_time	获取时间

### 1.3.22 订单表(order\_info)

字段名	字段说明
id	编号
consignee	收货人
consignee_tel	收件人电话
total_amount	总金额
order_status	订单状态
user_id	用户 id
payment_way	付款方式
delivery_address	送货地址
order_comment	订单备注
out_trade_no	订单交易编号 (第三方支付用)
trade_body	订单描述(第三方支付用)
create_time	创建时间
operate_time	操作时间
expire_time	失效时间
process_status	进度状态
tracking_no	物流单编号
parent_order_id	父订单编号
img_url	图片路径

province_id	地区
activity_reduce_amount	促销金额
coupon_reduce_amount	优惠金额
original_total_amount	原价金额
feight_fee	运费
feight_fee_reduce	运费减免
refundable_time	可退款日期（签收后 30 天）

### 1.3.23 退单表（order\_refund\_info）

字段名	字段说明
id	编号
user_id	用户 id
order_id	订单 id
sku_id	skuid
refund_type	退款类型
refund_num	退货件数
refund_amount	退款金额
refund_reason_type	原因类型
refund_reason_txt	原因内容
refund_status	退款状态（0：待审批 1：已退款）
create_time	创建时间

### 1.3.24 订单状态流水表（order\_status\_log）

字段名	字段说明
id	编号
order_id	订单编号
order_status	订单状态
operate_time	操作时间

### 1.3.25 支付表（payment\_info）

字段名	字段说明
id	编号
out_trade_no	对外业务编号
order_id	订单编号
user_id	
payment_type	支付类型（微信 支付宝）
trade_no	交易编号
total_amount	支付金额
subject	交易内容
payment_status	支付状态
create_time	创建时间
callback_time	回调时间
callback_content	回调信息

### 1.3.26 退款表 (refund\_payment)

字段名	字段说明
id	编号
out_trade_no	对外业务编号
order_id	订单编号
sku_id	商品 sku_id
payment_type	支付类型 (微信 支付宝)
trade_no	交易编号
total_amount	退款金额
subject	交易内容
refund_status	退款状态
create_time	创建时间
callback_time	回调时间
callback_content	回调信息

### 1.3.27 SKU 平台属性表 (sku\_attr\_value)

字段名	字段说明
id	编号
attr_id	属性 id (冗余)
value_id	属性值 id
sku_id	skuid
attr_name	属性名称
value_name	属性值名称

### 1.3.28 SKU 信息表 (sku\_info)

字段名	字段说明
id	库存 id(itemID)
spu_id	商品 id
price	价格
sku_name	sku 名称
sku_desc	商品规格描述
weight	重量
tm_id	品牌(冗余)
category3_id	三级分类 id (冗余)
sku_default_img	默认显示图片(冗余)
is_sale	是否销售 (1: 是 0: 否)
create_time	创建时间

### 1.3.29 SKU 销售属性表 (sku\_sale\_attr\_value)

字段名	字段说明
id	id
sku_id	库存单元 id

spu_id	spu_id (冗余)
sale_attr_value_id	销售属性值 id
sale_attr_id	销售属性 id
sale_attr_name	销售属性值名称
sale_attr_value_name	销售属性值名称

### 1.3.30 SPU 信息表 (spu\_info)

字段名	字段说明
id	商品 id
spu_name	商品名称
description	商品描述(后台简述)
category3_id	三级分类 id
tm_id	品牌 id

### 1.3.31 SPU 销售属性表 (spu\_sale\_attr)

字段名	字段说明
id	编号 (业务中无关联)
spu_id	商品 id
base_sale_attr_id	销售属性 id
sale_attr_name	销售属性名称 (冗余)

### 1.3.32 SPU 销售属性值表 (spu\_sale\_attr\_value)

字段名	字段说明
id	销售属性值编号
spu_id	商品 id
base_sale_attr_id	销售属性 id
sale_attr_value_name	销售属性值名称
sale_attr_name	销售属性名称 (冗余)

### 1.3.33 用户地址表 (user\_address)

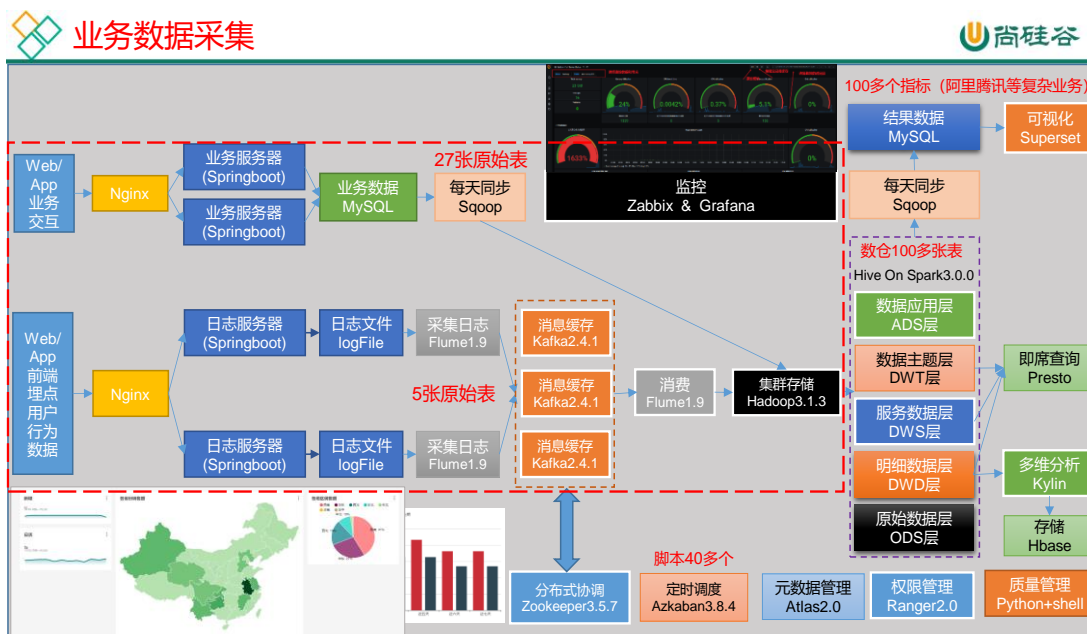
字段名	字段说明
id	编号
user_id	用户 id
province_id	省份 id
user_address	用户地址
consignee	收件人
phone_num	联系方式
is_default	是否是默认

### 1.3.34 用户信息表 (user\_info)

字段名	字段说明
id	编号
login_name	用户名称

nick_name	用户昵称
passwd	用户密码
name	用户姓名
phone_num	手机号
email	邮箱
head_img	头像
user_level	用户级别
birthday	用户生日
gender	性别 M 男,F 女
create_time	创建时间
operate_time	修改时间
status	状态

## 第 2 章 业务数据采集模块



### 2.1 MySQL 安装

#### 2.1.1 安装包准备

1) 将安装包和 JDBC 驱动上传到/opt/software，共计 6 个

```
01_mysql-community-common-5.7.16-1.el7.x86_64.rpm
02_mysql-community-libs-5.7.16-1.el7.x86_64.rpm
03_mysql-community-libs-compat-5.7.16-1.el7.x86_64.rpm
04_mysql-community-client-5.7.16-1.el7.x86_64.rpm
05_mysql-community-server-5.7.16-1.el7.x86_64.rpm
mysql-connector-java-5.1.27-bin.jar
```

2) 如果是虚拟机按照如下步骤执行

(1) 卸载自带的 Mysql-libs（如果之前安装过 MySQL，要全都卸载掉）

更多 Java - 大数据 - 前端 - python 人工智能资料下载，可百度访问：尚硅谷官网

```
[atguigu@hadoop102 software]$ rpm -qa | grep -i -E
mysql\|mariadb | xargs -n1 sudo rpm -e --nodeps
```

### 3) 如果是阿里云服务器按照如下步骤执行

说明：由于阿里云服务器安装的是 Linux 最小系统版，没有如下工具，所以需要安装。

(1) 卸载 MySQL 依赖，虽然机器上没有装 MySQL，但是这一步不可少

```
[atguigu@hadoop102 software]# sudo yum remove mysql-libs
```

(2) 下载依赖并安装

```
[atguigu@hadoop102 software]# sudo yum install libaio
[atguigu@hadoop102 software]# sudo yum -y install autoconf
```

## 2.1.2 安装 MySQL

### 1) 安装 MySQL 依赖

```
[atguigu@hadoop102 software]$ sudo rpm -ivh 01_mysql-community-
common-5.7.16-1.el7.x86_64.rpm
[atguigu@hadoop102 software]$ sudo rpm -ivh 02_mysql-community-
libs-5.7.16-1.el7.x86_64.rpm
[atguigu@hadoop102 software]$ sudo rpm -ivh 03_mysql-community-
libs-compatible-5.7.16-1.el7.x86_64.rpm
```

### 2) 安装 mysql-client

```
[atguigu@hadoop102 software]$ sudo rpm -ivh 04_mysql-community-
client-5.7.16-1.el7.x86_64.rpm
```

### 3) 安装 mysql-server

```
[atguigu@hadoop102 software]$ sudo rpm -ivh 05_mysql-community-
server-5.7.16-1.el7.x86_64.rpm
```

注意：如果报如下错误，这是由于 yum 安装了旧版本的 GPG keys 所造成，从 rpm 版

本 4.1 后，在安装或升级软件包时会自动检查软件包的签名。

```
warning:      05_mysql-community-server-5.7.16-1.el7.x86_64.rpm:
Header V3 DSA/SHA1 Signature, key ID 5072elf5: NOKEY
error: Failed dependencies:
libaio.so.1()(64bit) is needed by mysql-community-server-
5.7.16-1.el7.x86_64
```

解决办法

```
[atguigu@hadoop102 software]$ sudo rpm -ivh 05_mysql-community-
server-5.7.16-1.el7.x86_64.rpm --force --nodeps
```

### 4) 启动 MySQL

```
[atguigu@hadoop102 software]$ sudo systemctl start mysqld
```

### 5) 查看 MySQL 密码

```
[atguigu@hadoop102 software]$ sudo cat /var/log/mysqld.log |
grep password
```

## 2.1.3 配置 MySQL

配置只要是 root 用户 + 密码，在任何主机上都能登录 MySQL 数据库。

### 1) 用刚刚查到的密码进入 MySQL（如果报错，给密码加单引号）

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：尚硅谷官网

```
[atguigu@hadoop102 software]$ mysql -uroot -p'password'
```

- 2) 设置复杂密码（由于 MySQL 密码策略，此密码必须足够复杂）

```
mysql> set password=password("Qs23=zs32");
```

- 3) 更改 MySQL 密码策略

```
mysql> set global validate_password_length=4;  
mysql> set global validate_password_policy=0;
```

- 4) 设置简单好记的密码

```
mysql> set password=password("000000");
```

- 5) 进入 MySQL 库

```
mysql> use mysql
```

- 6) 查询 user 表

```
mysql> select user, host from user;
```

- 7) 修改 user 表，把 Host 表内容修改为%

```
mysql> update user set host="%" where user="root";
```

- 8) 刷新

```
mysql> flush privileges;
```

- 9) 退出

```
mysql> quit;
```

## 2.2 业务数据生成

### 2.2.1 连接 MySQL

通过 MySQL 可视化客户端连接数据库。

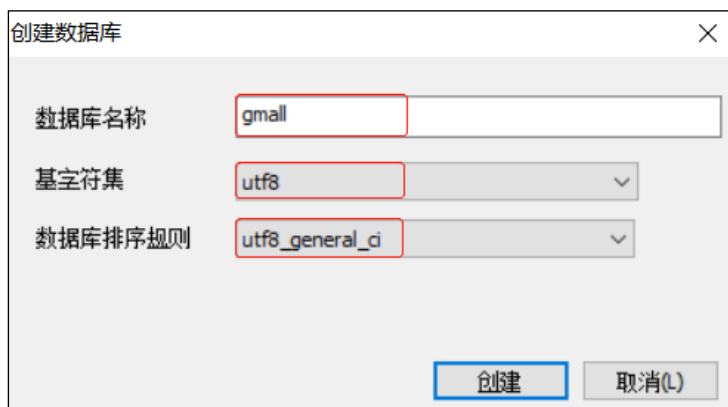


## 2.2.2 建表语句

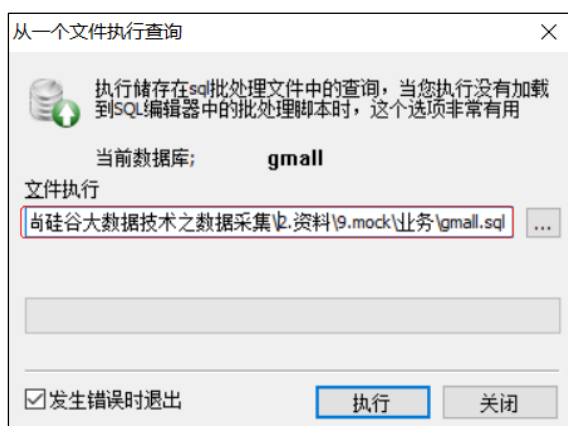
1) 通过 SQLyog 创建数据库



2) 设置数据库名称为 **gmall**，编码为 **utf-8**，排序规则为 **utf8\_general\_ci**



3) 导入数据库结构脚本 (**gmall.sql**)



注意：完成后，要记得右键，刷新一下对象浏览器，就可以看见数据库中的表了。

## 2.2.3 生成业务数据

1) 在 hadoop102 的 /opt/module/ 目录下创建 db\_log 文件夹

```
[atguigu@hadoop102 module]$ mkdir db_log/
```

2) 把 gmall2020-mock-db-2021-01-22.jar 和 application.properties 上传到 hadoop102 的

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：[尚硅谷官网](#)



/opt/module/db\_log 路径上。

### 3) 根据需求修改 application.properties 相关配置

```
logging.level.root=info

spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://hadoop102:3306/gmall?characterEncoding=utf-8&useSSL=false&serverTimezone=GMT%2B8
spring.datasource.username=root
spring.datasource.password=000000

logging.pattern.console=%m%n

mybatis-plus.global-config.db-config.field-strategy=not_null

#业务日期
mock.date=2020-06-14
#是否重置 注意：第一次执行必须设置为 1，后续不需要重置不用设置为 1
mock.clear=1
#是否重置用户 注意：第一次执行必须设置为 1，后续不需要重置不用设置为 1
mock.clear.user=1

#生成新用户数量
mock.user.count=100
#男性比例
mock.user.male-rate=20
#用户数据变化概率
mock.user.update-rate=20

#收藏取消比例
mock.favor.cancel-rate=10
#收藏数量
mock.favor.count=100

#每个用户添加购物车的概率
mock.cart.user-rate=50
#每次每个用户最多添加多少种商品进购物车
mock.cart.max-sku-count=8
#每个商品最多买几个
mock.cart.max-sku-num=3

#购物车来源 用户查询，商品推广，智能推荐，促销活动
mock.cart.source-type-rate=60:20:10:10

#用户下单比例
mock.order.user-rate=50
#用户从购物中购买商品比例
mock.order.sku-rate=50
#是否参加活动
mock.order.join-activity=1
#是否使用购物券
```

```
mock.order.use-coupon=1
#购物券领取人数
mock.coupon.user-count=100

#支付比例
mock.payment.rate=70
#支付方式 支付宝：微信：银联
mock.payment.payment-type=30:60:10

#评价比例 好：中：差：自动
mock.comment.appraise-rate=30:10:10:50

#退款原因比例：质量问题 商品描述与实际描述不一致 缺货 号码不合适 拍错 不想买了 其他
mock.refund.reason-rate=30:10:20:5:15:5:5
```

4) 并在该目录下执行，如下命令，生成 2020-06-14 日期数据：

```
[atguigu@hadoop102 db_log]$ java -jar gmall2020-mock-db-2021-01-22.jar
```

5) 查看 gmall 数据库，观察是否有 2020-06-14 的数据出现

## 2.2.4 业务数据建模

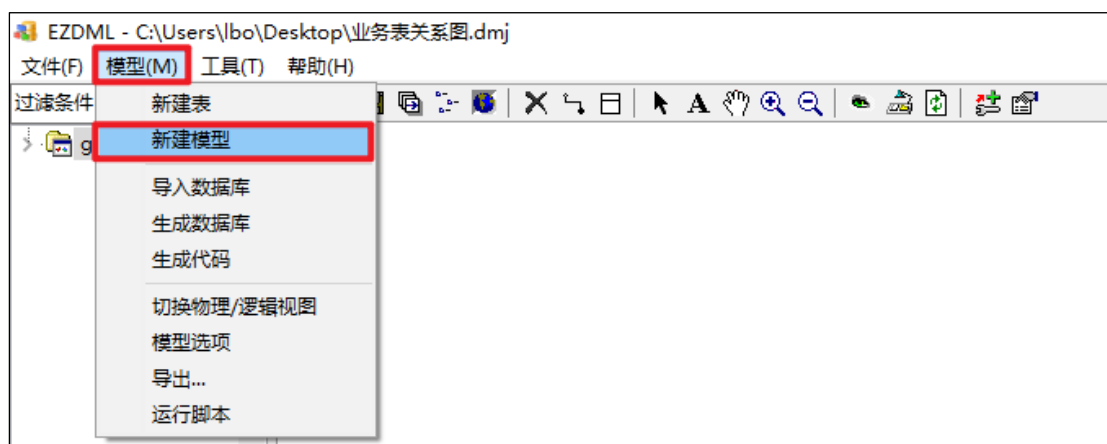
可借助 EZDML 这款数据库设计工具，来辅助我们梳理复杂的业务表关系。

1) 下载地址

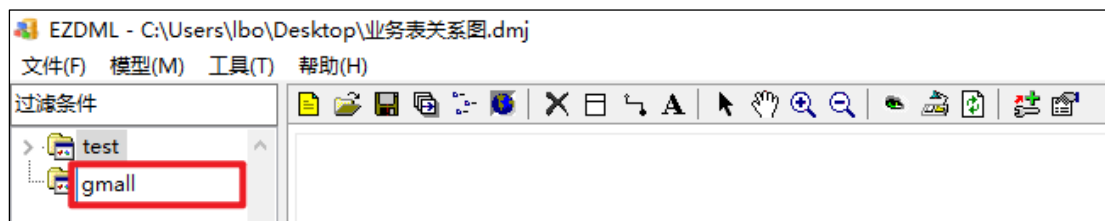
[http://www.ezdml.com/download\\_cn.html](http://www.ezdml.com/download_cn.html)

2) 使用说明

(1) 新建模型



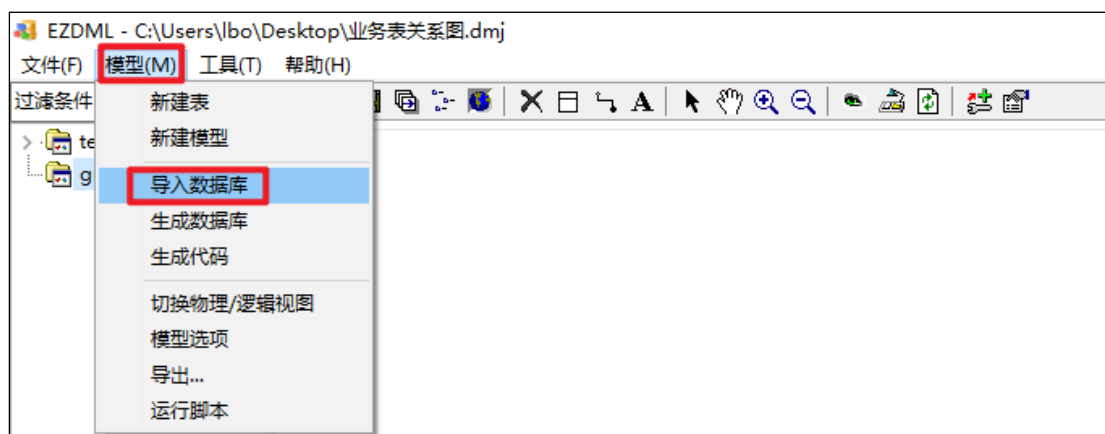
(2) 命名模型



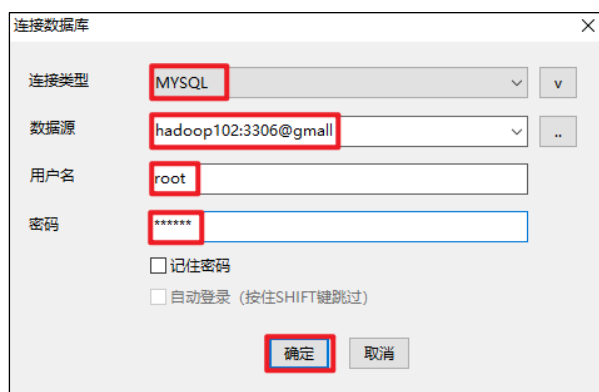
(3) 点击图标，选中模型



(4) 导入数据库



(5) 配置数据库连接



(6) 选择导入的表（标注红点的表不需要导入）

连接数据库 [MYSQL]hadoop102:3306@gmail

选择用户 **gmail**

过滤条件

选择对象

<input checked="" type="checkbox"/> activity_info	<input checked="" type="checkbox"/> comment_info	<input checked="" type="checkbox"/> sku_info
<input checked="" type="checkbox"/> activity_rule	<input checked="" type="checkbox"/> coupon_info	<input checked="" type="checkbox"/> sku_sale_attr_value
<input checked="" type="checkbox"/> activity_sku	<input checked="" type="checkbox"/> coupon_range	<input checked="" type="checkbox"/> spu_image
<input checked="" type="checkbox"/> base_attr_info	<input checked="" type="checkbox"/> coupon_use	<input checked="" type="checkbox"/> spu_info
<input checked="" type="checkbox"/> base_attr_value	<input checked="" type="checkbox"/> favor_info	<input checked="" type="checkbox"/> spu_poster
<input checked="" type="checkbox"/> base_category1	<input checked="" type="checkbox"/> financial_sku_cost	<input checked="" type="checkbox"/> spu_sale_attr
<input checked="" type="checkbox"/> base_category2	<input checked="" type="checkbox"/> order_detail	<input checked="" type="checkbox"/> spu_sale_attr_value
<input checked="" type="checkbox"/> base_category3	<input checked="" type="checkbox"/> order_detail_activity	<input checked="" type="checkbox"/> user_address
<input checked="" type="checkbox"/> base_category_view	<input checked="" type="checkbox"/> order_detail_coupon	<input checked="" type="checkbox"/> user_info
<input checked="" type="checkbox"/> base_dic	<input checked="" type="checkbox"/> order_info	<input checked="" type="checkbox"/> ware_info
<input checked="" type="checkbox"/> base_frontend_param	<input checked="" type="checkbox"/> order_refund_info	<input checked="" type="checkbox"/> ware_order_task
<input checked="" type="checkbox"/> base_province	<input checked="" type="checkbox"/> order_status_log	<input checked="" type="checkbox"/> ware_order_task_detail
<input checked="" type="checkbox"/> base_region	<input checked="" type="checkbox"/> payment_info	<input checked="" type="checkbox"/> ware_sku
<input checked="" type="checkbox"/> base_sale_attr	<input checked="" type="checkbox"/> refund_payment	
<input checked="" type="checkbox"/> base_trademark	<input checked="" type="checkbox"/> seckill_goods	
<input checked="" type="checkbox"/> cart_info	<input checked="" type="checkbox"/> sku_attr_value	
<input checked="" type="checkbox"/> cms_banner	<input checked="" type="checkbox"/> sku_image	

☐ 导入数据类型名称 ☐ 自动大小写 ☐ 注释转为逻辑名

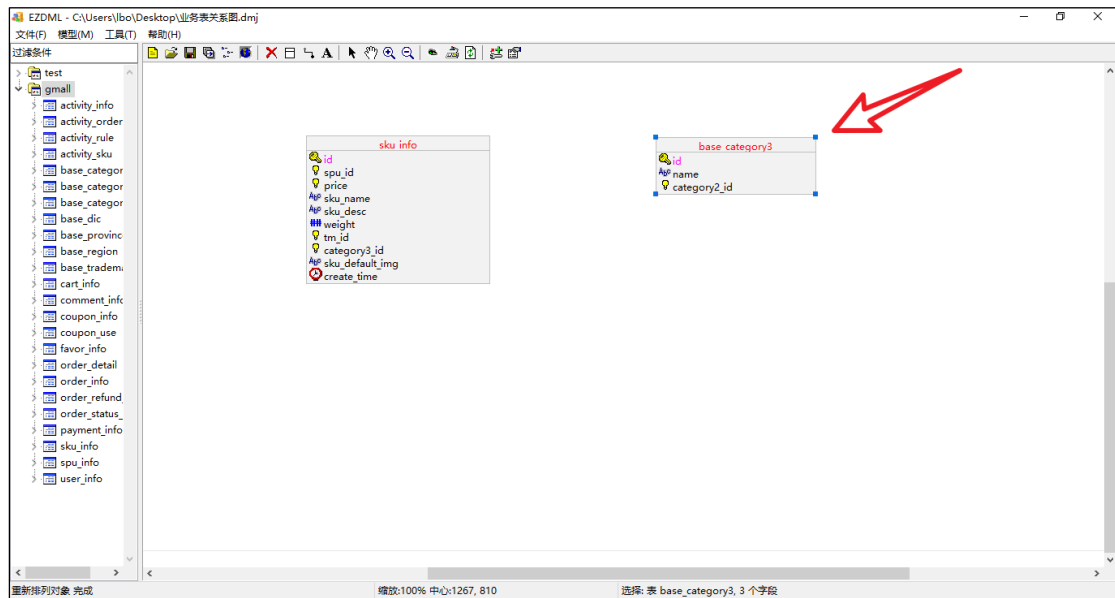
☒ 覆盖模型中的同名表

执行进度 0/0

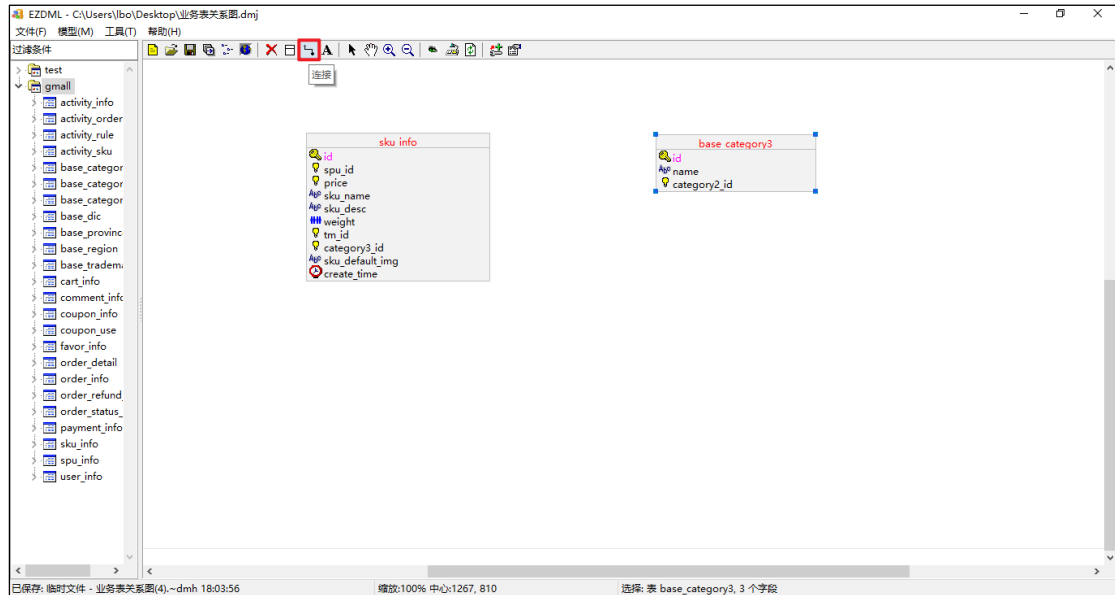
开始 关闭

## (7) 建立表关系

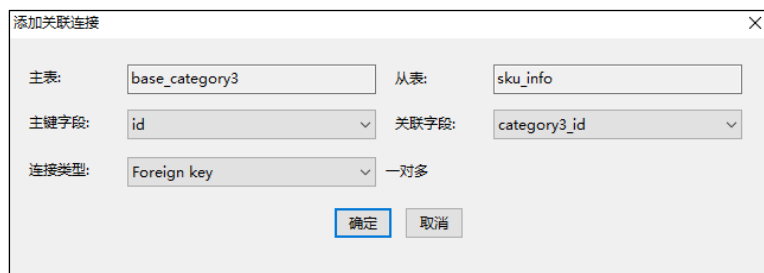
第一步：点击选中主表（主键所在的表）



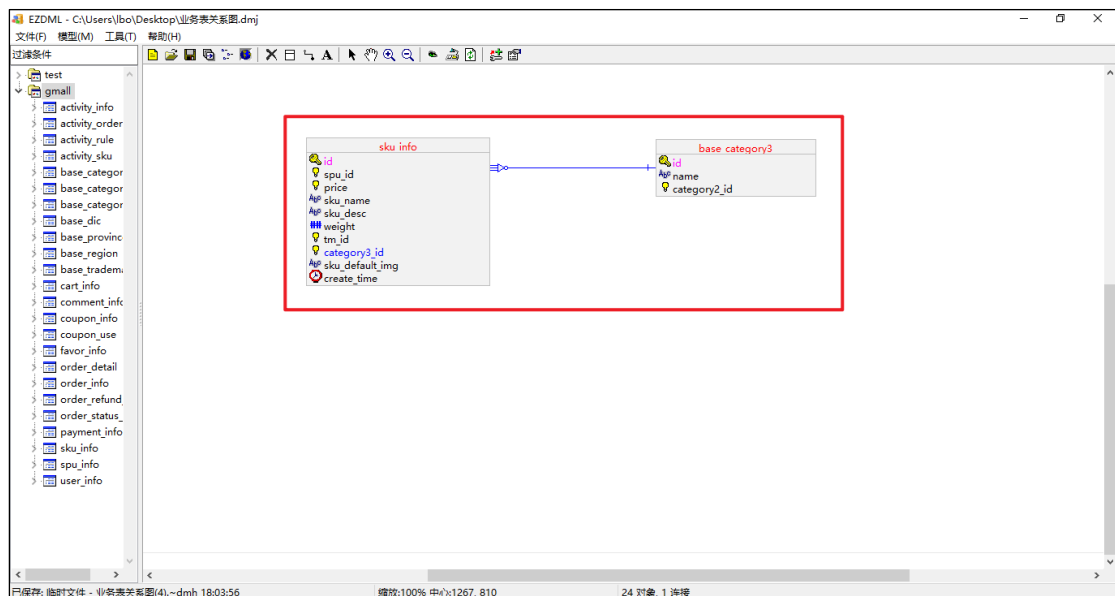
第二步：点击连接按钮



第三步：点击从表，配置连接条件



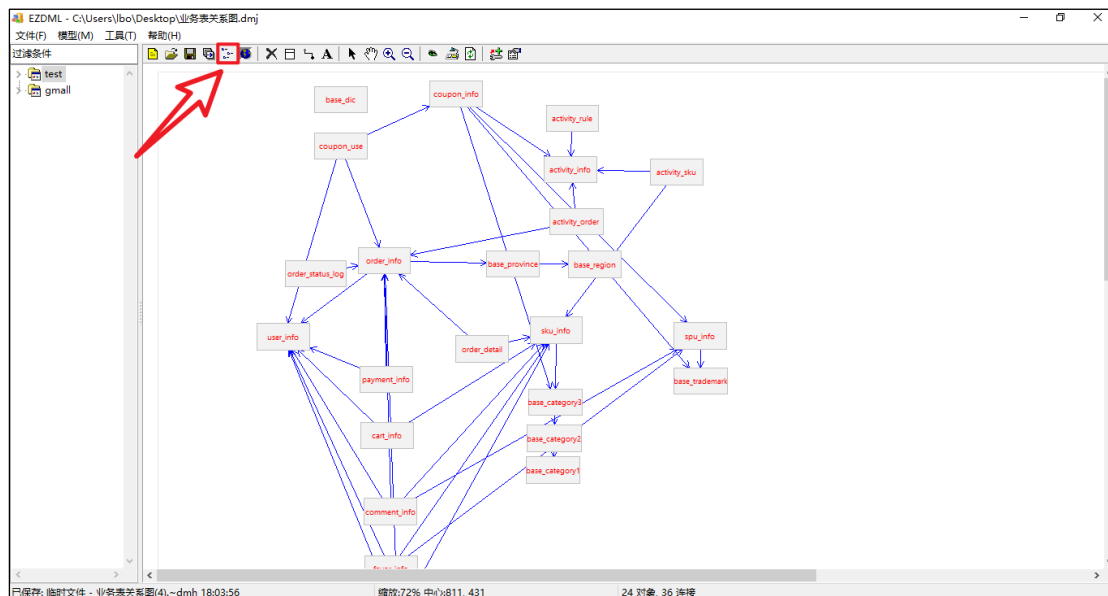
第四步：效果展示



### 3) 使用技巧

#### (1) 缩略图

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：[尚硅谷官网](#)



## (2) 热键

按住 shift 键，用鼠标点击表，进行多选，可实现批量移动

按住 ctrl 键，用鼠标圈选表，也可进行多选，实现批量移动

## 2.3 Sqoop 安装

### 2.3.1 下载并解压

- 1) sqoop 官网地址: <http://sqoop.apache.org>
- 2) 下载地址: <http://mirrors.hust.edu.cn/apache/sqoop/1.4.6/>
- 3) 上传安装包 sqoop-1.4.6.bin\_\_hadoop-2.0.4-alpha.tar.gz 到 hadoop102 的 /opt/software 路径中
- 4) 解压 sqoop 安装包到指定目录，如：

```
[atguigu@hadoop102 software]$ tar -zxvf sqoop-1.4.6.bin__hadoop-2.0.4-alpha.tar.gz -C /opt/module/
```

- 5) 解压 sqoop 安装包到指定目录，如：

```
[atguigu@hadoop102 module]$ mv sqoop-1.4.6.bin__hadoop-2.0.4-alpha/ sqoop
```

### 2.3.2 修改配置文件

- 1) 进入到 /opt/module/sqoop/conf 目录，重命名配置文件

```
[atguigu@hadoop102 conf]$ mv sqoop-env-template.sh sqoop-env.sh
```

- 2) 修改配置文件

```
[atguigu@hadoop102 conf]$ vim sqoop-env.sh
```

增加如下内容

```
export HADOOP_COMMON_HOME=/opt/module/hadoop-3.1.3
export HADOOP_MAPRED_HOME=/opt/module/hadoop-3.1.3
export HIVE_HOME=/opt/module/hive
```

更多 Java - 大数据 - 前端 - python 人工智能资料下载，可百度访问：尚硅谷官网

```
export ZOOKEEPER_HOME=/opt/module/zookeeper-3.5.7
export ZOOCFGDIR=/opt/module/zookeeper-3.5.7/conf
```

### 2.3.3 拷贝 JDBC 驱动

1) 将 mysql-connector-java-5.1.48.jar 上传到/opt/software 路径

2) 进入到/opt/software/路径，拷贝 jdbc 驱动到 sqoop 的 lib 目录下。

```
[atguigu@hadoop102 software]$ cp mysql-connector-java-5.1.48.jar /opt/module/sqoop/lib/
```

### 2.3.4 验证 Sqoop

(1) 我们可以通过某一个 command 来验证 sqoop 配置是否正确：

```
[atguigu@hadoop102 sqoop]$ bin/sqoop help
```

(2) 出现一些 Warning 警告（警告信息已省略），并伴随着帮助命令的输出：

```
Available commands:
  codegen                Generate code to interact with database
records
  create-hive-table      Import a table definition into Hive
  eval                  Evaluate a SQL statement and display the
results
  export                 Export an HDFS directory to a database table
  help                  List available commands
  import                Import a table from a database to HDFS
  import-all-tables     Import tables from a database to HDFS
  import-mainframe      Import datasets from a mainframe server to
HDFS
  job                   Work with saved jobs
  list-databases         List available databases on a server
  list-tables           List available tables in a database
  merge                 Merge results of incremental imports
  metastore             Run a standalone Sqoop metastore
  version               Display version information
```

### 2.3.5 测试 Sqoop 是否能够成功连接数据库

```
[atguigu@hadoop102 sqoop]$ bin/sqoop list-databases --connect
jdbc:mysql://hadoop102:3306/ --username root --password 000000
```

出现如下输出：

```
information_schema
metastore
mysql
oozie
performance_schema
```

### 2.3.6 Sqoop 基本使用

将 mysql 中 user\_info 表数据导入到 HDFS 的/test 路径

```
bin/sqoop import \
--connect jdbc:mysql://hadoop102:3306/gmall \
--username root \
--password 000000 \
--table user_info \
--columns id,login_name \
```

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：尚硅谷官网

```
--where "id>=10 and id<=30" \
--target-dir /test \
--delete-target-dir \
--fields-terminated-by '\t' \
--num-mappers 2 \
--split-by id
```

## 2.4 同步策略

数据同步策略的类型包括：全量同步、增量同步、新增及变化同步、特殊情况

- 全量表：存储完整的数据。
- 增量表：存储新增加的数据。
- 新增及变化表：存储新增加的数据和变化的数据。
- 特殊表：只需要存储一次。

### 2.4.1 全量同步策略

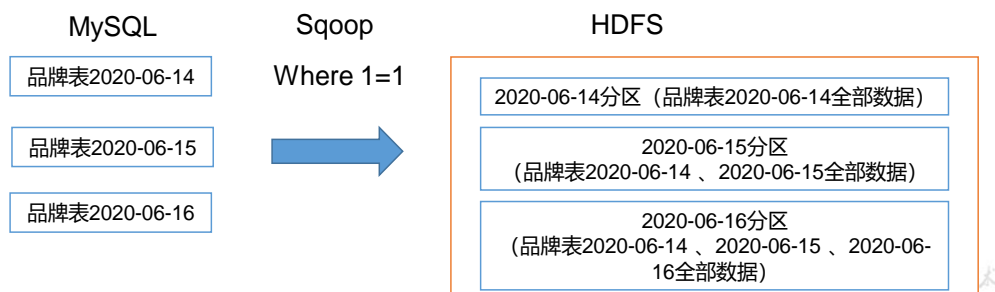
#### 全量同步策略



每日全量，就是每天存储一份**完整数据**，作为一个分区。

适用于表**数据量不大**，且每天既会有**新数据插入**，也会有**旧数据的修改**的场景。

例如：编码字典表、品牌表、商品三级分类、商品二级分类、商品一级分类、优惠规则表、活动表、活动参与商品表、加购表、商品收藏表、优惠券表、SKU商品表、SPU商品表



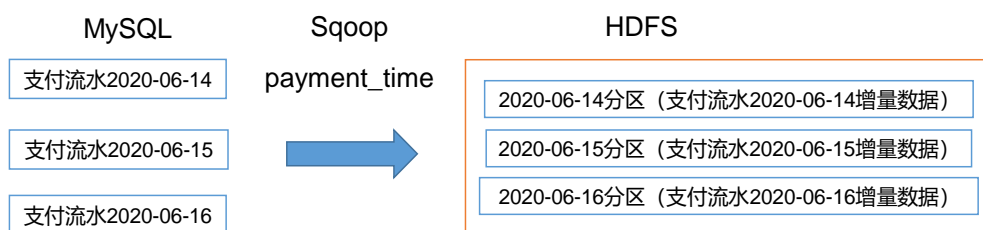


## 2.4.2 增量同步策略

### 增量同步策略

每日增量，就是每天存储一份增量数据，作为一个分区。

适用于表数据量大，且每天只有新数据插入的场景。例如：退单表、订单状态表、支付流水表、订单详情表、活动与订单关联表、商品评论表。



让天下没有难学的技术

## 2.4.3 新增及变化策略

### 新增及变化同步策略

每日新增及变化，就是存储创建时间和操作时间都是今天的数据。

适用场景为，表的数据量大，既会有新增，又会有变化。例如：用户表、订单表、优惠券领用表。

trade_body	create_time	operate_time
小米 (MI) 电视 55英寸曲面4K智能WiFi网络液晶电视机4S L55MS-AQ 小米电	2020-06-14 08:37:45	2020-06-15 08:39:34
北纯精制黄小米 (小黄米 月子米 小米粥 粗粮杂粮 大米伴侣) 2.18kg等1件	2020-06-14 08:37:45	2020-06-15 08:39:34
Apple iPhoneXSMAX (A2104) 256GB 深空灰色 移动联通电信4G手机 双卡双待	2020-06-14 08:37:45	2020-06-15 08:39:34
小米 (MI) 小米路由4 双千兆路由 无线家用穿墙1200M高速双频wifi 4	2020-06-14 08:37:45	2020-06-14 08:37:45
迪奥 (Dior) 烈艳蓝金唇膏/口红 珊瑚粉 ACTRICE 028号 3.5g等2件商品	2020-06-14 08:37:45	2020-06-14 08:37:45
北纯精制黄小米 (小黄米 月子米 小米粥 粗粮杂粮 大米伴侣) 2.18kg等3件	2020-06-14 08:37:45	2020-06-14 08:37:45
Apple iPhoneXSMAX (A2104) 256GB 深空灰色 移动联通电信4G手机 双卡双待	2020-06-14 08:37:45	2020-06-14 08:37:45
迪奥 (Dior) 烈艳蓝金唇膏/口红 珊瑚粉 ACTRICE 028号 3.5g等2件商品	2020-06-14 08:37:45	2020-06-14 08:37:45
荣耀10青春版 幻影渐变 2400万AI自拍 全网通版4GB+64GB 渐变蓝 移动联通	2020-06-14 08:37:45	2020-06-14 08:37:45
小米 (MI) 小米路由4 双千兆路由 无线家用穿墙1200M高速双频wifi 4	2020-06-15 08:39:33	2020-06-15 08:39:34
迪奥 (Dior) 烈艳蓝金唇膏/口红 珊瑚粉 ACTRICE 028号 3.5g等4件商品	2020-06-15 08:39:33	2020-06-15 08:39:34

让天下没有难学的技术

## 2.4.4 特殊策略

某些特殊的表，可不必遵循上述同步策略。例如某些不会发生变化的表（地区表，省份表，民族表）可以只存一份固定值。

## 2.5 业务数据导入 HDFS

### 2.5.1 分析表同步策略

在生产环境，个别小公司，为了简单处理，所有表全量导入。

更多 [Java - 大数据 - 前端 - python 人工智能资料下载](#)，可百度访问：尚硅谷官网

中大型公司，由于数据量比较大，还是严格按照同步策略导入数据。

## 分析表同步策略

数据量小/码表	数据量比较大+不变化	数据量大+变化	不变化
全量	增量	新增和变化	特殊
base_dic 编码字典表	activity_info 活动表	order_redund_info 退单表 (特殊)	coupon_use 优惠券领用表
base_trademark 品牌表	activity_rule 优惠规则表	order_status_log 订单状态表	user_info 用户表
base_category3 商品三级分类	cart_info 加购表 (特殊)	order_detail 订单详情表	order_info 订单表
base_category2 商品二级分类	favor_info 收藏表 (特殊)	order_detail_activity 订单明细活动关联表	payment_info 支付表
base_category1 商品一级分类	coupon_info 优惠券表	order_detail_coupon 订单明细优惠券关联表	refund_payment 退款表
sku_info SKU商品表	sku_attr_value SKU平台属性表	comment_info 商品评论表	
spu_info SPU商品表	sku_sale_attr_value SKU销售属性表		

让天下没有难学的技术

## 2.5.2 业务数据首日同步脚本

### 1) 脚本编写

(1) 在/home/atguigu/bin 目录下创建

```
[atguigu@hadoop102 bin]$ vim mysql_to_hdfs_init.sh
```

添加如下内容：

```
#!/bin/bash

APP=gmall
sqoop=/opt/module/sqoop/bin/sqoop

if [ -n "$2" ] ;then
    do_date=$2
else
    echo "请传入日期参数"
    exit
fi

import_data(){
    $sqoop import \
    --connect jdbc:mysql://hadoop102:3306/$APP \
    --username root \
    --password 000000 \
    --target-dir /origin_data/$APP/db/$1/$do_date \
    --delete-target-dir \
    --query "$2 where \$CONDITIONS" \
    --num-mappers 1 \
    --fields-terminated-by '\t' \
    --compress \
    --compression-codec lzop \
    --null-string '\N' \
    --null-non-string '\N'
}
```

更多 Java - 大数据 - 前端 - python 人工智能资料下载，可百度访问：尚硅谷官网

```
hadoop jar /opt/module/hadoop-3.1.3/share/hadoop/common/hadoop-  
lzo-0.4.20.jar com.hadoop.compression.lzo.DistributedLzoIndexer  
/origin_data/$APP/db/$1/$do_date  
}  
  
import_order_info(){  
    import_data order_info "select  
        id,  
        total_amount,  
        order_status,  
        user_id,  
        payment_way,  
        delivery_address,  
        out_trade_no,  
        create_time,  
        operate_time,  
        expire_time,  
        tracking_no,  
        province_id,  
        activity_reduce_amount,  
        coupon_reduce_amount,  
        original_total_amount,  
        feight_fee,  
        feight_fee_reduce  
    from order_info"  
}  
  
import_coupon_use(){  
    import_data coupon_use "select  
        id,  
        coupon_id,  
        user_id,  
        order_id,  
        coupon_status,  
        get_time,  
        using_time,  
        used_time,  
        expire_time  
    from coupon_use"  
}  
  
import_order_status_log(){  
    import_data order_status_log "select  
        id,  
        order_id,  
        order_status,  
        operate_time  
    from order_status_log"  
}  
  
import_user_info(){  
    import_data "user_info" "select  
        id,  
        login_name,  
        nick_name,  
        name,  
        phone_num,  
        email,
```

```
        user_level,
        birthday,
        gender,
        create_time,
        operate_time
    from user_info"
}

import_order_detail(){
    import_data order_detail "select
        id,
        order_id,
        sku_id,
        sku_name,
        order_price,
        sku_num,
        create_time,
        source_type,
        source_id,
        split_total_amount,
        split_activity_amount,
        split_coupon_amount
    from order_detail"
}

import_payment_info(){
    import_data "payment_info" "select
        id,
        out_trade_no,
        order_id,
        user_id,
        payment_type,
        trade_no,
        total_amount,
        subject,
        payment_status,
        create_time,
        callback_time
    from payment_info"
}

import_comment_info(){
    import_data comment_info "select
        id,
        user_id,
        sku_id,
        spu_id,
        order_id,
        appraise,
        create_time
    from comment_info"
}

import_order_refund_info(){
    import_data order_refund_info "select
        id,
        user_id,
        order_id,
```

```
        sku_id,
        refund_type,
        refund_num,
        refund_amount,
        refund_reason_type,
        refund_status,
        create_time
    from order_refund_info"
}

import_sku_info(){
    import_data sku_info "select
        id,
        spu_id,
        price,
        sku_name,
        sku_desc,
        weight,
        tm_id,
        category3_id,
        is_sale,
        create_time
    from sku_info"
}

import_base_category1(){
    import_data "base_category1" "select
        id,
        name
    from base_category1"
}

import_base_category2(){
    import_data "base_category2" "select
        id,
        name,
        category1_id
    from base_category2"
}

import_base_category3(){
    import_data "base_category3" "select
        id,
        name,
        category2_id
    from base_category3"
}

import_base_province(){
    import_data base_province "select
        id,
        name,
        region_id,
        area_code,
        iso_code,
        iso_3166_2
    from base_province"
}
```

```
import_base_region(){
    import_data base_region "select
                                id,
                                region_name
                                from base_region"
}

import_base_trademark(){
    import_data base_trademark "select
                                id,
                                tm_name
                                from base_trademark"
}

import_spu_info(){
    import_data spu_info "select
                            id,
                            spu_name,
                            category3_id,
                            tm_id
                            from spu_info"
}

import_favor_info(){
    import_data favor_info "select
                            id,
                            user_id,
                            sku_id,
                            spu_id,
                            is_cancel,
                            create_time,
                            cancel_time
                            from favor_info"
}

import_cart_info(){
    import_data cart_info "select
                            id,
                            user_id,
                            sku_id,
                            cart_price,
                            sku_num,
                            sku_name,
                            create_time,
                            operate_time,
                            is_ordered,
                            order_time,
                            source_type,
                            source_id
                            from cart_info"
}

import_coupon_info(){
    import_data coupon_info "select
                            id,
                            coupon_name,
                            coupon_type,
```

```
        condition_amount,
        condition_num,
        activity_id,
        benefit_amount,
        benefit_discount,
        create_time,
        range_type,
        limit_num,
        taken_count,
        start_time,
        end_time,
        operate_time,
        expire_time
    from coupon_info"
}

import_activity_info(){
    import_data activity_info "select
        id,
        activity_name,
        activity_type,
        start_time,
        end_time,
        create_time
    from activity_info"
}

import_activity_rule(){
    import_data activity_rule "select
        id,
        activity_id,
        activity_type,
        condition_amount,
        condition_num,
        benefit_amount,
        benefit_discount,
        benefit_level
    from activity_rule"
}

import_base_dic(){
    import_data base_dic "select
        dic_code,
        dic_name,
        parent_code,
        create_time,
        operate_time
    from base_dic"
}

import_order_detail_activity(){
    import_data order_detail_activity "select
        id,
        order_id,
        order_detail_id,
        activity_id,
```

```
activity_rule_id,
                                sku_id,
                                create_time
                                from
order_detail_activity"
}

import_order_detail_coupon(){
    import_data order_detail_coupon "select
                                id,
order_id,
order_detail_id,
                                coupon_id,
coupon_use_id,
                                sku_id,
                                create_time
                                from
order_detail_coupon"
}

import_refund_payment(){
    import_data refund_payment "select
                                id,
                                out_trade_no,
                                order_id,
                                sku_id,
                                payment_type,
                                trade_no,
                                total_amount,
                                subject,
                                refund_status,
                                create_time,
                                callback_time
                                from refund_payment"
}

import_sku_attr_value(){
    import_data sku_attr_value "select
                                id,
                                attr_id,
                                value_id,
                                sku_id,
                                attr_name,
                                value_name
                                from sku_attr_value"
}

import_sku_sale_attr_value(){
    import_data sku_sale_attr_value "select
                                id,
```



```

                                sku_id,
                                spu_id,

sale_attr_value_id,                                sale_attr_id,
                                                    sale_attr_name,

sale_attr_value_name

                                                    from
sku_sale_attr_value"
}

case $1 in
    "order_info")
        import_order_info
    ;;
    "base_category1")
        import_base_category1
    ;;
    "base_category2")
        import_base_category2
    ;;
    "base_category3")
        import_base_category3
    ;;
    "order_detail")
        import_order_detail
    ;;
    "sku_info")
        import_sku_info
    ;;
    "user_info")
        import_user_info
    ;;
    "payment_info")
        import_payment_info
    ;;
    "base_province")
        import_base_province
    ;;
    "base_region")
        import_base_region
    ;;
    "base_trademark")
        import_base_trademark
    ;;
    "activity_info")
        import_activity_info
    ;;
    "cart_info")
        import_cart_info
    ;;
    "comment_info")
        import_comment_info
    ;;
    "coupon_info")
        import_coupon_info
    ;;
;;
```

```
"coupon_use")
    import_coupon_use
;;
"favor_info")
    import_favor_info
;;
"order_refund_info")
    import_order_refund_info
;;
"order_status_log")
    import_order_status_log
;;
"spu_info")
    import_spu_info
;;
"activity_rule")
    import_activity_rule
;;
"base_dic")
    import_base_dic
;;
"order_detail_activity")
    import_order_detail_activity
;;
"order_detail_coupon")
    import_order_detail_coupon
;;
"refund_payment")
    import_refund_payment
;;
"sku_attr_value")
    import_sku_attr_value
;;
"sku_sale_attr_value")
    import_sku_sale_attr_value
;;
"all")
    import_base_category1
    import_base_category2
    import_base_category3
    import_order_info
    import_order_detail
    import_sku_info
    import_user_info
    import_payment_info
    import_base_region
    import_base_province
    import_base_trademark
    import_activity_info
    import_cart_info
    import_comment_info
    import_coupon_use
    import_coupon_info
    import_favor_info
    import_order_refund_info
    import_order_status_log
    import_spu_info
    import_activity_rule
```

```
import_base_dic
import_order_detail_activity
import_order_detail_coupon
import_refund_payment
import_sku_attr_value
import_sku_sale_attr_value
;;
esac
```

说明 1:

**[ -n 变量值 ]** 判断变量的值，是否为空

-- 变量的值，非空，返回 true

-- 变量的值，为空，返回 false

说明 2:

查看 date 命令的使用，[atguigu@hadoop102 ~]\$ **date --help**

(2) 增加脚本执行权限

```
[atguigu@hadoop102 bin]$ chmod +x mysql_to_hdfs_init.sh
```

## 2) 脚本使用

```
[atguigu@hadoop102 bin]$ mysql_to_hdfs_init.sh all 2020-06-14
```

## 2.5.3 业务数据每日同步脚本

### 1) 脚本编写

(1) 在/home/atguigu/bin 目录下创建

```
[atguigu@hadoop102 bin]$ vim mysql_to_hdfs.sh
```

添加如下内容:

```
#!/bin/bash

APP=gmall
sqoop=/opt/module/sqoop/bin/sqoop

if [ -n "$2" ] ;then
    do_date=$2
else
    do_date=`date -d '-1 day' +%F`
fi

import_data(){
    $sqoop import \
    --connect jdbc:mysql://hadoop102:3306/$APP \
    --username root \
    --password 000000 \
    --target-dir /origin_data/$APP/db/$1/$do_date \
    --delete-target-dir \
    --query "$2 and \$CONDITIONS" \
    --num-mappers 1 \
    --fields-terminated-by '\t' \
    --compress \
    --compression-codec lzop \
```

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：尚硅谷官网

```
--null-string '\\N' \
--null-non-string '\\N'

hadoop jar /opt/module/hadoop-3.1.3/share/hadoop/common/hadoop-
lzo-0.4.20.jar com.hadoop.compression.lzo.DistributedLzoIndexer
/origin_data/$APP/db/$1/$do_date
}

import_order_info(){
import_data order_info "select
                        id,
                        total_amount,
                        order_status,
                        user_id,
                        payment_way,
                        delivery_address,
                        out_trade_no,
                        create_time,
                        operate_time,
                        expire_time,
                        tracking_no,
                        province_id,
                        activity_reduce_amount,
                        coupon_reduce_amount,
                        original_total_amount,
                        feight_fee,
                        feight_fee_reduce
                    from order_info
                    where
(date_format(create_time,'%Y-%m-%d')='$do_date'
or
date_format(operate_time,'%Y-%m-%d')='$do_date') "
}

import_coupon_use(){
import_data coupon_use "select
                        id,
                        coupon_id,
                        user_id,
                        order_id,
                        coupon_status,
                        get_time,
                        using_time,
                        used_time,
                        expire_time
                    from coupon_use
                    where
(date_format(get_time,'%Y-%m-%d')='$do_date'
or
date_format(using_time,'%Y-%m-%d')='$do_date'
or
date_format(used_time,'%Y-%m-%d')='$do_date'
or
date_format(expire_time,'%Y-%m-%d')='$do_date') "
}

import_order_status_log(){
import_data order_status_log "select
```

```
        id,
        order_id,
        order_status,
        operate_time
    from order_status_log
    where
date_format(operate_time,'%Y-%m-%d')='$do_date'"
}

import_user_info(){
    import_data "user_info" "select
        id,
        login_name,
        nick_name,
        name,
        phone_num,
        email,
        user_level,
        birthday,
        gender,
        create_time,
        operate_time
    from user_info
    where
    (DATE_FORMAT(create_time,'%Y-%m-%d')='$do_date'
    or
    DATE_FORMAT(operate_time,'%Y-%m-%d')='$do_date')"
}

import_order_detail(){
    import_data order_detail "select
        id,
        order_id,
        sku_id,
        sku_name,
        order_price,
        sku_num,
        create_time,
        source_type,
        source_id,
        split_total_amount,
        split_activity_amount,
        split_coupon_amount
    from order_detail
    where
    DATE_FORMAT(create_time,'%Y-%m-%d')='$do_date'"
}

import_payment_info(){
    import_data "payment_info" "select
        id,
        out_trade_no,
        order_id,
        user_id,
        payment_type,
        trade_no,
        total_amount,
        subject,
```

```
        payment_status,
        create_time,
        callback_time
    from payment_info
    where
    (DATE_FORMAT(create_time,'%Y-%m-%d')='$do_date'
    or
    DATE_FORMAT(callback_time,'%Y-%m-%d')='$do_date') "
}

import_comment_info(){
    import_data comment_info "select
        id,
        user_id,
        sku_id,
        spu_id,
        order_id,
        appraise,
        create_time
    from comment_info
    where
    date_format(create_time,'%Y-%m-%d')='$do_date'"
}

import_order_refund_info(){
    import_data order_refund_info "select
        id,
        user_id,
        order_id,
        sku_id,
        refund_type,
        refund_num,
        refund_amount,
        refund_reason_type,
        refund_status,
        create_time
    from order_refund_info
    where
    date_format(create_time,'%Y-%m-%d')='$do_date'"
}

import_sku_info(){
    import_data sku_info "select
        id,
        spu_id,
        price,
        sku_name,
        sku_desc,
        weight,
        tm_id,
        category3_id,
        is_sale,
        create_time
    from sku_info where 1=1"
}

import_base_category1(){
    import_data "base_category1" "select
```

```
        id,
        name
    from base_category1 where 1=1"
}

import_base_category2(){
    import_data "base_category2" "select
        id,
        name,
        category1_id
    from base_category2 where 1=1"
}

import_base_category3(){
    import_data "base_category3" "select
        id,
        name,
        category2_id
    from base_category3 where 1=1"
}

import_base_province(){
    import_data base_province "select
        id,
        name,
        region_id,
        area_code,
        iso_code,
        iso_3166_2
    from base_province
    where 1=1"
}

import_base_region(){
    import_data base_region "select
        id,
        region_name
    from base_region
    where 1=1"
}

import_base_trademark(){
    import_data base_trademark "select
        id,
        tm_name
    from base_trademark
    where 1=1"
}

import_spu_info(){
    import_data spu_info "select
        id,
        spu_name,
        category3_id,
        tm_id
    from spu_info
    where 1=1"
}
```

```
import_favor_info(){
    import_data favor_info "select
        id,
        user_id,
        sku_id,
        spu_id,
        is_cancel,
        create_time,
        cancel_time
    from favor_info
    where 1=1"
}

import_cart_info(){
    import_data cart_info "select
        id,
        user_id,
        sku_id,
        cart_price,
        sku_num,
        sku_name,
        create_time,
        operate_time,
        is_ordered,
        order_time,
        source_type,
        source_id
    from cart_info
    where 1=1"
}

import_coupon_info(){
    import_data coupon_info "select
        id,
        coupon_name,
        coupon_type,
        condition_amount,
        condition_num,
        activity_id,
        benefit_amount,
        benefit_discount,
        create_time,
        range_type,
        limit_num,
        taken_count,
        start_time,
        end_time,
        operate_time,
        expire_time
    from coupon_info
    where 1=1"
}

import_activity_info(){
    import_data activity_info "select
        id,
        activity_name,
```



```
        activity_type,
        start_time,
        end_time,
        create_time
    from activity_info
    where 1=1"
}

import_activity_rule(){
    import_data activity_rule "select
        id,
        activity_id,
        activity_type,
        condition_amount,
        condition_num,
        benefit_amount,
        benefit_discount,
        benefit_level
    from activity_rule
    where 1=1"
}

import_base_dic(){
    import_data base_dic "select
        dic_code,
        dic_name,
        parent_code,
        create_time,
        operate_time
    from base_dic
    where 1=1"
}

import_order_detail_activity(){
    import_data order_detail_activity "select
        id,
        order_id,
        order_detail_id,
        activity_id,
        activity_rule_id,
        sku_id,
        create_time
    from
    order_detail_activity
    where
    date_format(create_time,'%Y-%m-%d')='$do_date'"
}

import_order_detail_coupon(){
    import_data order_detail_coupon "select
        id,
        order_id,
```

```
order_detail_id,
                                                    coupon_id,
coupon_use_id,
                                                    sku_id,
                                                    create_time
                                                    from
order_detail_coupon
                                                    where
date_format(create_time,'%Y-%m-%d')='$do_date'
}

import_refund_payment(){
    import_data refund_payment "select
                                                    id,
                                                    out_trade_no,
                                                    order_id,
                                                    sku_id,
                                                    payment_type,
                                                    trade_no,
                                                    total_amount,
                                                    subject,
                                                    refund_status,
                                                    create_time,
                                                    callback_time
                                                    from refund_payment
                                                    where
            (DATE_FORMAT(create_time,'%Y-%m-%d')='$do_date'
            or
            DATE_FORMAT(callback_time,'%Y-%m-%d')='$do_date')"
    }

import_sku_attr_value(){
    import_data sku_attr_value "select
                                                    id,
                                                    attr_id,
                                                    value_id,
                                                    sku_id,
                                                    attr_name,
                                                    value_name
                                                    from sku_attr_value
                                                    where 1=1"
    }

import_sku_sale_attr_value(){
    import_data sku_sale_attr_value "select
                                                    id,
                                                    sku_id,
                                                    spu_id,
sale_attr_value_id,
                                                    sale_attr_id,
                                                    sale_attr_name,
sale_attr_value_name
```

```

                                from
sku_sale_attr_value
                                where 1=1 "
}

case $1 in
    "order_info")
        import_order_info
    ;;
    "base_category1")
        import_base_category1
    ;;
    "base_category2")
        import_base_category2
    ;;
    "base_category3")
        import_base_category3
    ;;
    "order_detail")
        import_order_detail
    ;;
    "sku_info")
        import_sku_info
    ;;
    "user_info")
        import_user_info
    ;;
    "payment_info")
        import_payment_info
    ;;
    "base_province")
        import_base_province
    ;;
    "activity_info")
        import_activity_info
    ;;
    "cart_info")
        import_cart_info
    ;;
    "comment_info")
        import_comment_info
    ;;
    "coupon_info")
        import_coupon_info
    ;;
    "coupon_use")
        import_coupon_use
    ;;
    "favor_info")
        import_favor_info
    ;;
    "order_refund_info")
        import_order_refund_info
    ;;
    "order_status_log")
        import_order_status_log
    ;;
    "spu_info")
```

```
import_spu_info
;;
"activity_rule")
    import_activity_rule
;;
"base_dic")
    import_base_dic
;;
"order_detail_activity")
    import_order_detail_activity
;;
"order_detail_coupon")
    import_order_detail_coupon
;;
"refund_payment")
    import_refund_payment
;;
"sku_attr_value")
    import_sku_attr_value
;;
"sku_sale_attr_value")
    import_sku_sale_attr_value
;;
"all")
    import_base_category1
    import_base_category2
    import_base_category3
    import_order_info
    import_order_detail
    import_sku_info
    import_user_info
    import_payment_info
    import_base_trademark
    import_activity_info
    import_cart_info
    import_comment_info
    import_coupon_use
    import_coupon_info
    import_favor_info
    import_order_refund_info
    import_order_status_log
    import_spu_info
    import_activity_rule
    import_base_dic
    import_order_detail_activity
    import_order_detail_coupon
    import_refund_payment
    import_sku_attr_value
    import_sku_sale_attr_value
;;
esac
```

(2) 增加脚本执行权限

```
[atguigu@hadoop102 bin]$ chmod +x mysql_to_hdfs.sh
```

## 2) 脚本使用

```
[atguigu@hadoop102 bin]$ mysql_to_hdfs.sh all 2020-06-15
```

## 2.5.4 项目经验

Hive 中的 Null 在底层是以 “\N” 来存储，而 MySQL 中的 Null 在底层就是 Null，为了保证数据两端的一致性。在导出数据时采用--input-null-string 和--input-null-non-string 两个参数。导入数据时采用--null-string 和--null-non-string。

## 第 3 章 数据环境准备

### 3.1 Hive 安装部署

(1) 把 apache-hive-3.1.2-bin.tar.gz 上传到 Linux 的/opt/software 目录下

(2) 解压 apache-hive-3.1.2-bin.tar.gz 到/opt/module/目录下

```
[atguigu@hadoop102 software]$ tar -zxvf /opt/software/apache-hive-3.1.2-bin.tar.gz -C /opt/module/
```

(3) 修改 apache-hive-3.1.2-bin.tar.gz 的名称为 hive

```
[atguigu@hadoop102 software]$ mv /opt/module/apache-hive-3.1.2-bin/ /opt/module/hive
```

(4) 修改/etc/profile.d/my\_env.sh，添加环境变量

```
[atguigu@hadoop102 software]$ sudo vim /etc/profile.d/my_env.sh
```

添加内容

```
#HIVE_HOME
export HIVE_HOME=/opt/module/hive
export PATH=$PATH:$HIVE_HOME/bin
```

重启 Xshell 对话框或者 source 一下 /etc/profile.d/my\_env.sh 文件，使环境变量生效

```
[atguigu@hadoop102 software]$ source /etc/profile.d/my_env.sh
```

(5) 解决日志 Jar 包冲突，进入/opt/module/hive/lib 目录

```
[atguigu@hadoop102 lib]$ mv log4j-slf4j-impl-2.10.0.jar log4j-slf4j-impl-2.10.0.jar.bak
```

### 3.2 Hive 元数据配置到 MySQL

#### 3.2.1 拷贝驱动

将 MySQL 的 JDBC 驱动拷贝到 Hive 的 lib 目录下

```
[atguigu@hadoop102 lib]$ cp /opt/software/mysql-connector-java-5.1.27.jar /opt/module/hive/lib/
```

#### 3.2.2 配置 Metastore 到 MySQL

(1) 在\$HIVE\_HOME/conf 目录下新建 hive-site.xml 文件

```
[atguigu@hadoop102 conf]$ vim hive-site.xml
```

(2) 添加如下内容

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
```

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：尚硅谷官网

```
<configuration>
  <property>
    <name>javax.jdo.option.ConnectionURL</name>

<value>jdbc:mysql://hadoop102:3306/metastore?useSSL=false</value>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>com.mysql.jdbc.Driver</value>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionUserName</name>
    <value>root</value>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <value>000000</value>
  </property>

  <property>
    <name>hive.metastore.warehouse.dir</name>
    <value>/user/hive/warehouse</value>
  </property>

  <property>
    <name>hive.metastore.schema.validation</name>
    <value>false</value>
  </property>

  <property>
    <name>hive.server2.thrift.port</name>
    <value>10000</value>
  </property>

  <property>
    <name>hive.server2.thrift.bind.host</name>
    <value>hadoop102</value>
  </property>

  <property>
    <name>hive.metastore.event.db.notification.api.auth</name>
    <value>false</value>
  </property>

  <property>
    <name>hive.cli.print.header</name>
    <value>true</value>
  </property>

  <property>
    <name>hive.cli.print.current.db</name>
    <value>true</value>
  </property>
```

```
</configuration>
```

## 3.3 启动 Hive

### 3.3.1 初始化元数据库

(1) 登陆 MySQL

```
[atguigu@hadoop102 conf]$ mysql -uroot -p000000
```

(2) 新建 Hive 元数据库

```
mysql> create database metastore;  
mysql> quit;
```

(3) 初始化 Hive 元数据库

```
[atguigu@hadoop102 conf]$ schematool -initSchema -dbType mysql  
-verbose
```

### 3.3.2 启动 Hive 客户端

(1) 启动 Hive 客户端

```
[atguigu@hadoop102 hive]$ bin/hive
```

(2) 查看一下数据库

```
hive (default)> show databases;  
OK  
database_name  
default
```