

Web 项目1-day05-课堂笔记

Web 项目1-day05-课堂笔记

商品模块-商品数据表设计

- 1.1 【理解掌握】SPU 和 SKU 概念
- 1.2 【理解掌握】首页广告数据表分析
- 1.3 【理解掌握】商品相关数据表分析

商品模块-相关测试数据

- 2.1 【简单了解】FastDFS 文件存储系统
- 2.2 【掌握操作】Docker软件基本使用
- 2.3 【掌握操作】FastDFS 文件上传和访问

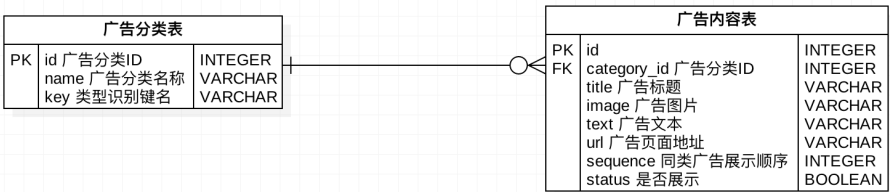
商品模块-商品数据表设计

1.1 【理解掌握】SPU 和 SKU 概念

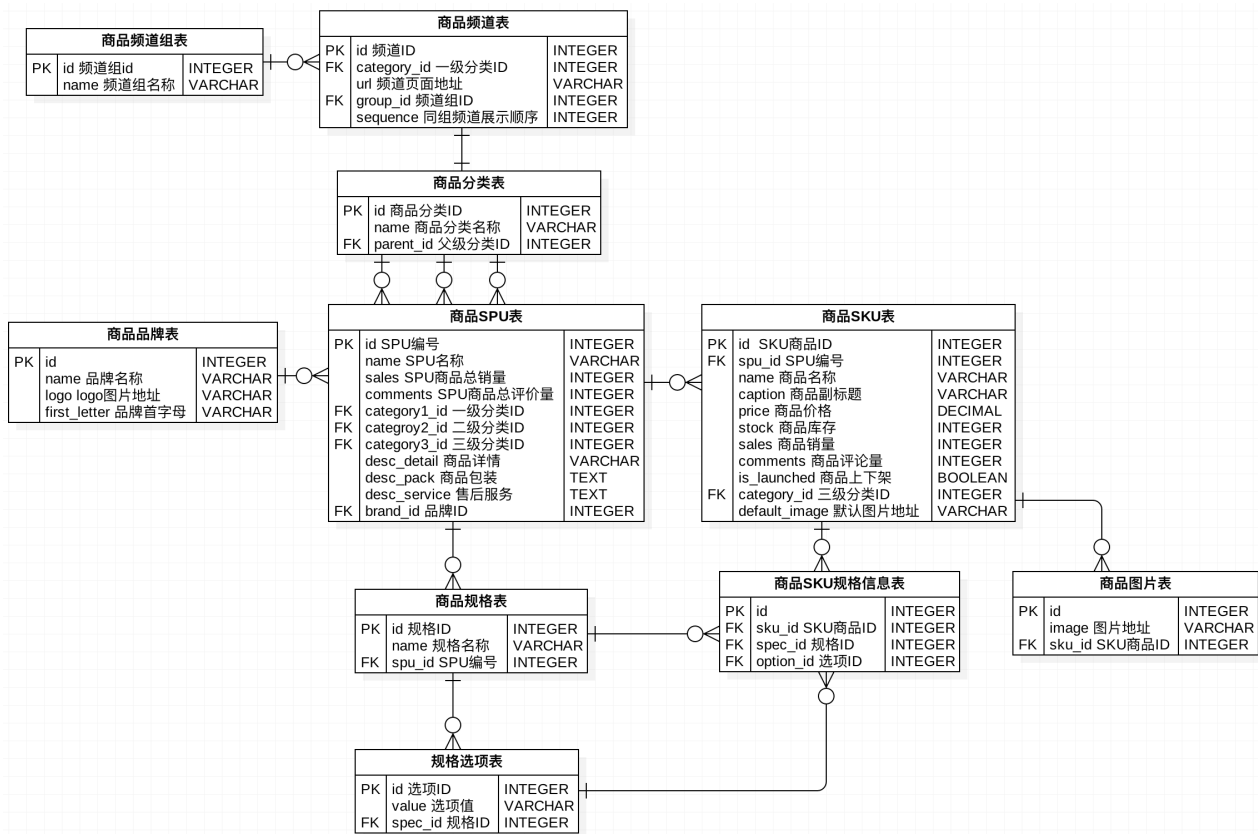
SPU：属性值、特性相同的商品的统称。比如：iPhone X 就是一个 SPU。

SKU：是指具体到属性值、规格的一款特定商品。比如：iPhone X 全网通 黑色 256G 就是一个 SKU。

1.2 【理解掌握】首页广告数据表分析



1.3 【理解掌握】商品相关数据表分析



商品模块-相关测试数据

2.1 【简单了解】FastDFS 文件存储系统

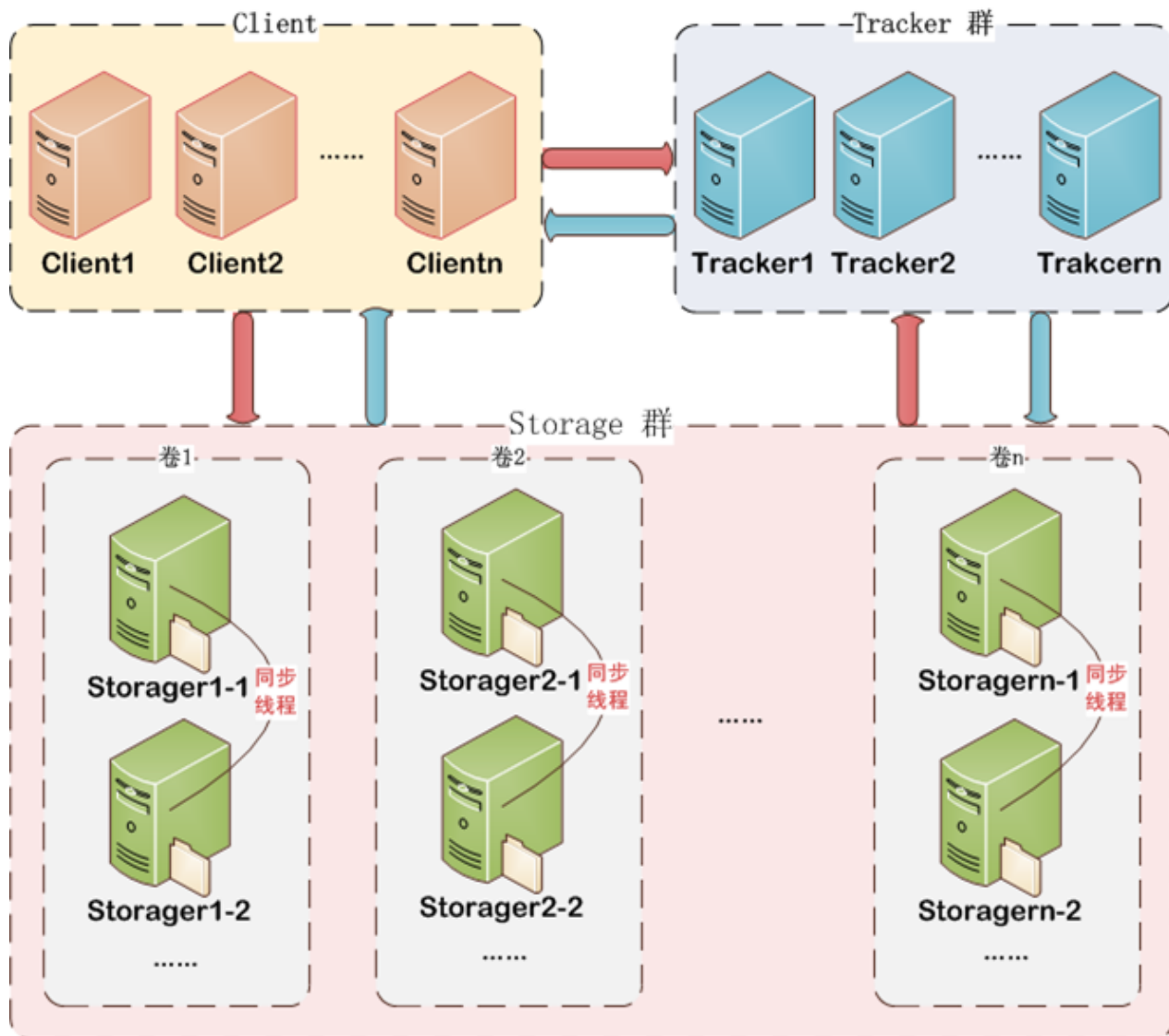
FastDFS 是一个用 c 语言编写的一款开源的轻量级分布式文件系统。

其功能主要包括：

- 1) 文件存储、文件访问（文件上传、文件下载）、文件同步等，解决了大容量存储和负载均衡的问题。特别适合以文件为载体的在线服务，如相册网站、视频网站等等。
- 2) 为互联网量身定制，充分考虑了冗余备份、负载均衡、线性扩容等机制，并注重高可用、高性能等指标。
- 3) 可以帮助我们搭建一套高性能的文件服务器集群，并提供文件上传、下载等服务。

架构说明：

FastDFS 文件系统架构中，主要包括三大功能组件：Client、Tracker Server 和 Storage Server。其中 Client 请求Tracker 进行文件上传、下载，Tracker 再调度 Storage 完成文件上传和下载。



组件功能说明：

- 1) **Client**: 客户端，业务请求的发起方，通过专有接口，使用 TCP/IP 协议与 Tracker 或 Storage 进行数据交互。FastDFS 提供了 upload、download、delete 等接口供客户端使用。
- 2) **Tracker server**: 跟踪服务器，主要做调度工作，起负载均衡的作用。在内存中记录集群中所有 Storage 组和 Storage 服务器的状态信息，是客户端和 Storage 服务器交互的枢纽。
- 3) **Storage server**: 存储服务器（存储节点或数据服务器），文件和文件属性都保存到存储服务器上。Storage Server 直接利用 OS 的文件系统调用管理文件。Storage 群中的横向可以扩容，纵向可以备份。

2.2 【掌握操作】 Docker软件基本使用

软件简介：

Docker是一个开源的容器引擎，简单来说：Docker就是一种快速解决业务稳定环境的一种技术手段，可以快速构建项目运行所依赖的环境。

基本概念：

- **Docker镜像(image)**: 它是一个只读的文件，就类似于我们安装操作系统时候所需要的那个iso光盘镜像，通过运行这个镜像来完成各种应用的部署。这里的镜像就是一个能被docker运行起来的一

个程序。

- Docker容器(container): 容器就类似于我们运行起来的一个操作系统, 而且这个操作系统启动了某些服务。这里的容器指的是运行起来的一个Docker镜像。
- Docker仓库(repository): 仓库就类似于我们在网上搜索操作系统光盘的一个镜像站。这里的仓库指的是Docker镜像存储的地方。

镜像管理命令[熟练掌握]:

- 1) 搜索镜像: `docker search [image_name]`
- 2) 获取镜像: `docker pull [image_name]`
- 3) 查看本地镜像: `docker images`
- 4) 查看镜像历史: `docker history [image_name]`
- 5) 镜像重命名: `docker tag [old_image]:[old_version] [new_image]:[new_version]`
- 6) 删除镜像: `docker rmi [image_id]` 或 `docker rmi [image_name:image_version]`
- 7) 导出镜像: `docker save -o [包文件] [镜像]`
- 8) 导入镜像: `docker load < [image.tar_name]`

容器管理命令[熟练常握]:

- 1) 查看容器: `docker ps` 或 `docker ps -a`
- 2) 创建容器: `docker run <参数, 可选> [docker_image] [执行的命令]`
- 3) 启动容器: `docker start [container_id]` 或 `docker start [container_name]`
- 4) 关闭容器: `docker stop [container_id]` 或 `docker stop [container_name]`
- 5) 进入正在运行的容器:
`docker exec -it [container_name] /bin/bash` 或 `docker exec -it [container_id] /bin/bash`
- 6) 删除容器: `docker rm [container_id]` 或 `docker rm [container_name]`
- 7) 基于容器创建镜像:
`docker commit -m '说明信息' -a "作者信息" [container_id] [new_image:tag]`
或
`docker commit -m '说明信息' -a "作者信息" [container_name] [new_image:tag]`
- 8) 查看容器运行过程产生的日志: `docker logs [container_id]` 或 `docker logs [container_name]`
- 9) 查看容器详细信息: `docker inspect [container_id]` 或 `docker inspect [container_name]`

2.3 【常握操作】FastDFS 文件上传和访问

fdfs-client扩展包使用:

```
# 导入扩展包中的 Fdfs_client 类
from fdfs_client.client import Fdfs_client

# 创建一个 Fdfs_client 类的实例对象
# 注意：添加的路径应该为 client.conf 配置文件的绝对路径：
client =
Fdfs_client('/Users/smart/Desktop/meiduo/meiduo_mall/meiduo_mall/utils/fastdfs
/client.conf')

# 使用刚刚创建的 client 对象调用方法，测试上传文件
# 注意：这里 /Users/smart/Desktop/images/lufei.jpg 为老师电脑的文件路径，需要换成自己的
ret = client.upload_by_filename('/Users/smart/Desktop/images/lufei.jpg')
```

FastDFS 上传文件的访问：

