大数据项目之数仓

版本：V1.0

# 第1章 数据仓库概念

数据仓库是为企业所有级别的决策制定过程，提供所有类型数据支持的战略集合。

数据仓库是出于分析报告和决策支持目的而创建的，为需要业务智能的企业，提供指导业务流程改进、监控时间、成本、质量以及控制。

# 第2章 项目需求及架构设计

## 2.1 项目需求分析

一、数据采集平台搭建

二、实现数据仓库分层的搭建

三、实现数据清洗、聚合、计算等操作

四、统计各指标，如统计通过各地址跳转注册的用户人数、统计各平台的用户人数、统计支付金额topN的用户

## 2.2 项目框架

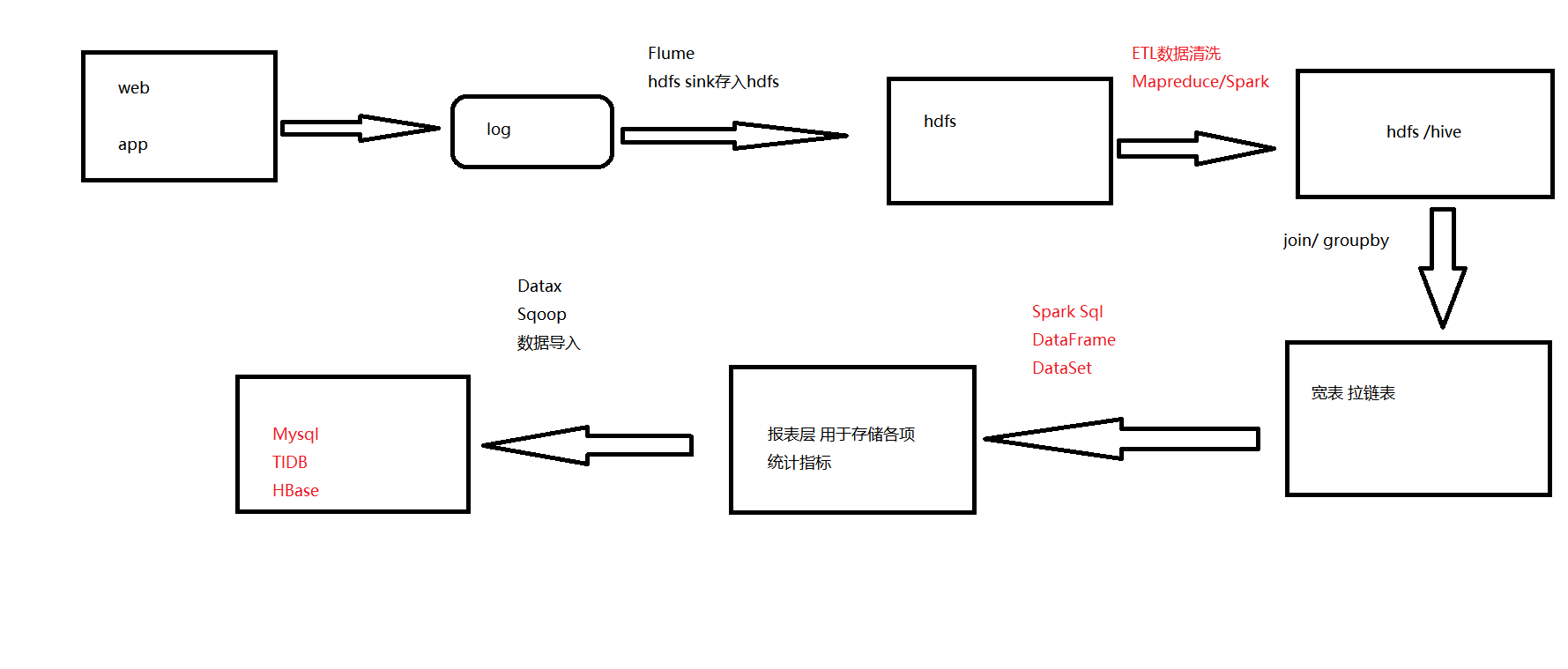
### 2.2.1 技术选型

一、数据存储：Hdfs

二、数据处理：Hive、Spark

三、任务调度：Azkaban

### 2.2.2 流程设计



### 2.2.3 框架版本选型



# 需求

## 3.0原始数据格式及字段含义

1.baseadlog 广告基础表原始json数据

{

"adid": "0",

"adname": "注册弹窗广告0",

"dn": "webA"

}

adid ：基础广告表广告id adname:广告详情名称 dn:网站分区(hive 里的website网站分区字段)

1. basewebsitelog 网站基础表原始json数据

{

"createtime": "2000-01-01",

"creator": "admin",

"delete": "0",

"dn": "webC",

"siteid": "2",

"sitename": "114",

"siteurl": "www.114.com/webC"

}

siteid:网站id sitename:网站名称 siteurl:网站地址 dn:网站分区（hive 里的website网站分区字段）

1. memberRegtype 用户跳转地址注册表

{

"appkey": "-",

"appregurl": "http:www.webA.com/product/register/index.html",

"bdp\_uuid": "-",

"createtime": "2015-05-11",

"dn": "webA",

"domain": "-",

"isranreg": "-",

"regsource": "4",

"uid": "0",

"websiteid": "0"

}

appregurl:注册时所跳转的地址

regsource:所属平台 1.PC 2.MOBILE 3.APP 4.WECHAT

uid:用户id

webisteid:对应basewebsitelog 下的siteid网站

dn：网站分区（hive 里的website网站分区字段）

1. pcentermempaymoneylog 用户支付金额表

{

"dn": "webA",

"paymoney": "162.54",

"siteid": "1",

"uid": "4376695",

"vip\_id": "0"

}

dn：网站分区（hive 里的website网站分区字段）

paymoney：支付金额

siteid:网站id对应 对应basewebsitelog 下的siteid网站

uid：用户id

vip\_id：对应pcentermemviplevellog vip\_id

dn：网站分区(hive里的website网站分区字段)

1. pcentermemviplevellog用户vip等级基础表

{

"discountval": "-",

"dn": "webA",

"end\_time": "2019-01-01",

"last\_modify\_time": "2019-01-01",

"max\_free": "-",

"min\_free": "-",

"next\_level": "-",

"operator": "update",

"start\_time": "2015-02-07",

"vip\_id": "2",

"vip\_level": "银卡"

}

dn:网站分区（hive 里的website网站分区字段）

vip\_id:vip\_id

vip\_level:vip级别名称

1. memberlog 用户基本信息表

{

"ad\_id": "0",

"birthday": "1981-08-14",

"dn": "webA",

"email": "test@126.com",

"fullname": "王69239",

"iconurl": "-",

"lastlogin": "-",

"mailaddr": "-",

"memberlevel": "6",

"password": "123456",

"paymoney": "-",

"phone": "13711235451",

"qq": "10000",

"register": "2016-08-15",

"regupdatetime": "-",

"uid": "69239",

"unitname": "-",

"userip": "123.235.75.48",

"zipcode": "-"

}

ad\_id:广告id dn:网站分区（hive里对应website分区字段） fullname:用户名

memberlevel:用户级别 password:密码 paymoney:支付金额（统计金额不用用户表的金额）

phone:手机号 uid:用户id

其余字段为非统计项 直接使用默认值存储即可

## 3.1数据分层

在hadoop集群上创建 odl目录



在hive里分别建立三个库，bdl、idl、adl 分别用于存储etl清洗后的数据、宽表和拉链表数据、各报表层统计指标数据。

create database bdl;

create database idl;

create database adl;

各层级对应关系 odl ->ods 存放原始数据

idl ->dwd 结构与原始表结构保持一致，对ods层数据进行清洗

bdl ->dws 以dwd为基础进行轻度汇总

adl ->ads 报表层，为各种统计报表提供数据

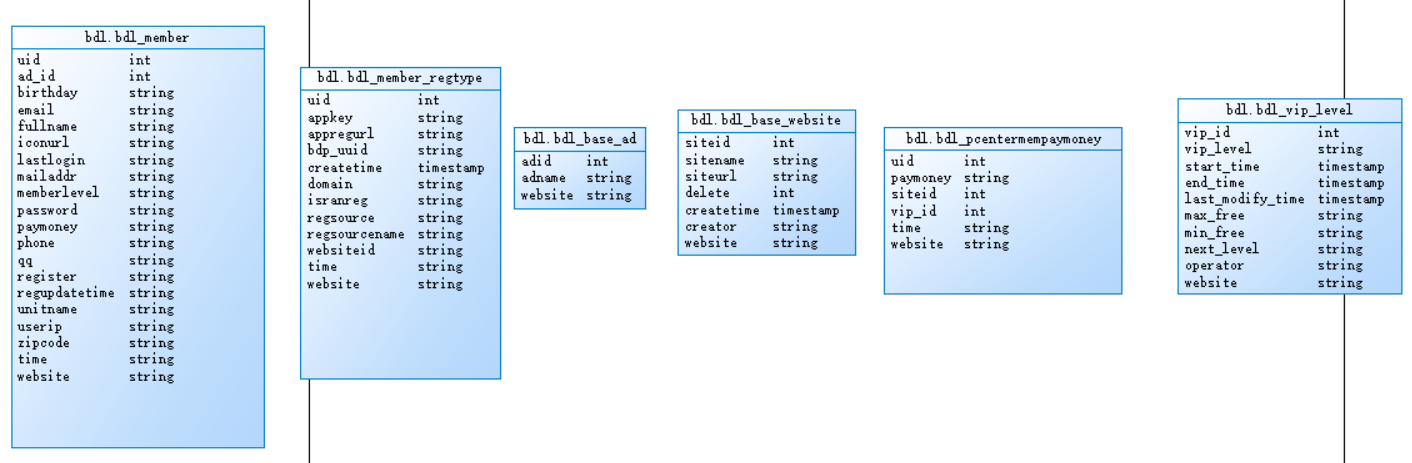
各层建表语句：



表模型：

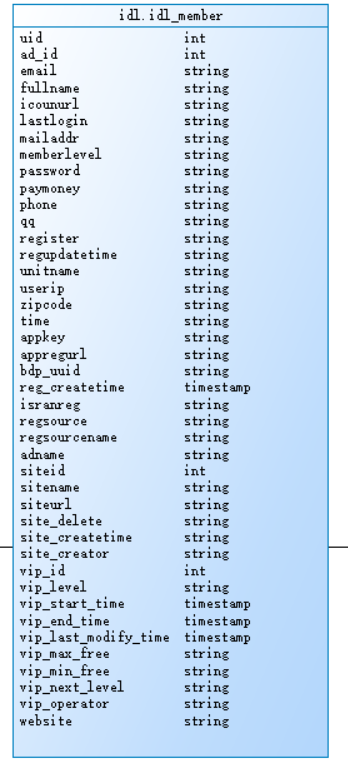


bdl层 6张基础表

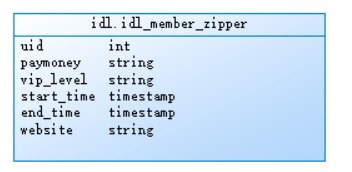


idl层 宽表和拉链表

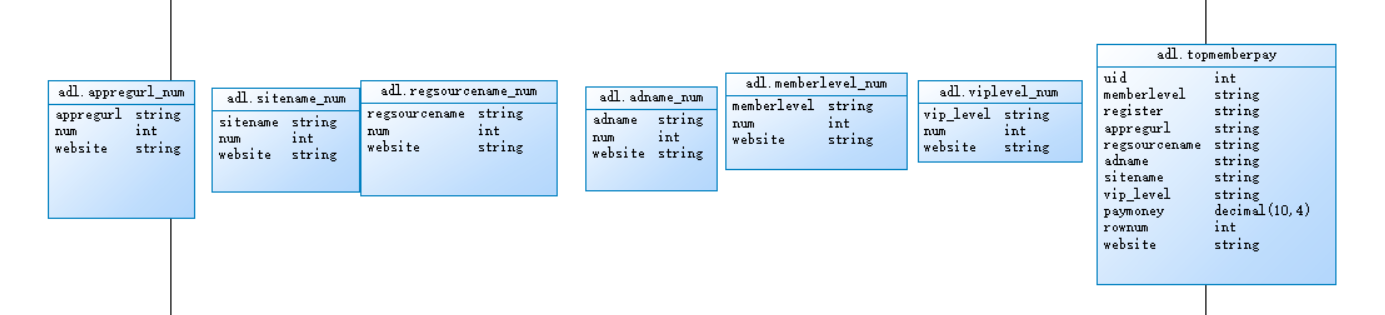
宽表



拉链表

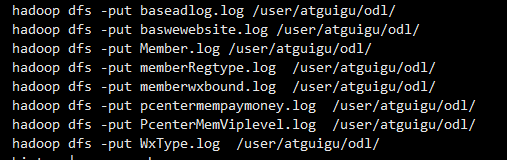


报表层各统计表



## 3.2模拟数据采集上传数据

模拟数据采集 将日志文件数据直接上传到hadoop集群上，



## 3.2 ETL清洗

需求1：必须使用Spark进行数据清洗，对用户名、手机号、密码进行脱敏处理，并使用Spark将数据导入到bdl层hive表中

清洗规则 用户名：王XX 手机号：137\*\*\*\*\*789 密码直接替换成\*\*\*\*\*\*

所有有time分区字段的表 格式化当前日期为yyyyMMdd作为分区字段

## 3.3基于bdl层表合成idl层的宽表

需求2：对bdl层的6张表进行合并，生成一张宽表，先使用Spark Sql实现。有时间的同学需要使用DataFrame api实现功能,并对join进行优化。

## 3.4拉链表

需求3：针对idl层宽表的支付金额（paymoney）和vip等级(vip\_level)这两个会变动的字段生成一张拉链表，需要一天进行一次更新

## 3.5报表层各指标统计

需求4：使用Spark DataFrame Api统计通过各注册跳转地址(appregurl)进行注册的用户数,有时间的再写Spark Sql

需求5：使用Spark DataFrame Api统计各所属网站（sitename）的用户数,有时间的再写Spark Sql

需求6：使用Spark DataFrame Api统计各所属平台的（regsourcename）用户数,有时间的再写Spark Sql

需求7：使用Spark DataFrame Api统计通过各广告跳转（adname）的用户数,有时间的再写Spark Sql

需求8：使用Spark DataFrame Api统计各用户级别（memberlevel）的用户数,有时间的再写Spark Sql

需求9：使用Spark DataFrame Api统计各分区网站、用户级别下(website、memberlevel)的top3用户,有时间的再写Spark Sql

# 第4章 思考

（1）第一层表哪些用overwrite合适，哪些用append合适

（2）数据过滤后，重组成需要的数据进行插入表的时候如何控制分区个数，即如何解决小文件过多问题。

（3）各基础表在进行join合成宽表时该用什么join，为什么

（4）合成宽表时一个用户会对应多条明细支付金额数据，如何合并

（5）分区的场景和作用，为什么需要分区