|  |  |
| --- | --- |
| 项目名称:Spark农产品分析系统 | 密级Confidentiality level |
| 设计说明书 | 内部公开 |
| 产品版本Product version | Total pages 共 18页 |
| 1.2 |

**Spark农产品分析系统 设计说明书**

|  |  |  |  |
| --- | --- | --- | --- |
| 拟制人： | 沈旗 | 时间： | 2022.06.09 |
| 评审人： |  | 时间： | 2022.06.09 |
| 批准人： |  | 时间： | 2022.06.09 |

**修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **日期** | **修订内容** | **修订人** |
| 1.0 | 2022/06/09 | 初稿 | 沈旗 |
| 1.1 | 2022/06/09 | 部分内容的修改 | 沈旗 |
| 1.2 | 2022/06/12 | 数据库部分修改 | 沈旗 |
|  |  |  |  |

**目录**

[1 研发背景 5](#_Toc500515855)

[2 阅读对象 5](#_Toc500515856)

[3 参考资料 5](#_Toc500515857)

[4 术语、缩略语 5](#_Toc500515858)

[5 概要设计 5](#_Toc500515859)

[5.1 需求概述 5](#_Toc500515860)

[5.2 技术架构与选型 6](#_Toc500515861)

[5.3 系统功能整体流程图 6](#_Toc500515861)

[5.4 系统数据流图 6](#_Toc500515861)

[5.5 项目架构流程图 6](#_Toc500515862)

[6 具体实现 6](#_Toc500515864)

[6.1 实现方法 6](#_Toc500515865)

[6.1.1 RDD 6](#_Toc500515866)

[6.1.2 累加器 6](#_Toc500515867)

[6.1.3 广播变量 6](#_Toc500515868)

[6.1.4 数据倾斜 6](#_Toc500515869)

[7 数据库设计 7](#_Toc500515873)

[7.1 数据库设计综述 7](#_Toc500515874)

[7.2 数据库物理结构设计 7](#_Toc500515876)

[7.3 数据字典 7](#_Toc500515878)

[8 项目实用性分析 7](#_Toc500515879)

[9 附录 7](#_Toc500515880)

# 研发背景

近年来，我们农产品市场运行呈现平稳的态势，供需基本平衡。总体呈现如下特点：一是供应充足；二是消费低增长；三是价格低增长；四是进口量增加；五是库存充裕。展望未来，我国农业现代化建设稳步发展，农产品消费结构转型加快，预计农产品市场价格将保持小幅上涨态势。面对错综复杂的国际形势和国内经济下行压力，我国农产品市场供应总体充足，基本满足消费升级需求，市场供需基本平衡。我国的粮食生产形势持续向好，粮食总产量边续12年增加，年均增长2.2%；油菜籽、花生等油料产量略有增加；猪、牛、羊等存栏稳定，肉类产量总体平稳。

# 阅读对象

实训老师，组内相关技术人员

# 参考资料

Rdd编程，数据库使用介绍

# 术语、缩略语

|  |  |
| --- | --- |
| 术语、缩略语 | 说明 |
| 数据库(Database) | 数据库(Database)是按照数据结构来组织、存储和管理数据的仓库，它产生于距今六十多年前，随着信息技术和市场的发展，特别是二十世纪九十年代以后，数据管理不再仅仅是存储和管理数据，而转变成用户所需要的各种数据管理的方式。数据库有很多种类型，从最简单的存储有各种数据的表格到能够进行海量数据存储的大型数据库系统都在各个方面得到了广泛的应用。 |
| Davinci(可视化) | 围绕 View（数据视图）与 Widget（可视化组件）两个核心概念设计View 是数据的结构化形态，一切逻辑/权限/服务等相关都是从 View 展开。Widget 是数据的可视化形态，一切展示/交互/引导等都是从 Widget 展开。作为数据的两种不同形态，二者相辅相成，让用户拥有一致的体验和认识 |
| HDFS | Hadoop分布式文件系统(HDFS)被设计成适合运行在通用硬件(commodity hardware)上的分布式文件系统。它和现有的分布式文件系统有很多共同点。但同时，它和其他的分布式文件系统的区别也是很明显的。HDFS是一个高度容错性的系统，适合部署在廉价的机器上。HDFS能提供高吞吐量的数据访问，非常适合大规模数据集上的应用。HDFS放宽了一部分POSIX约束，来实现流式读取文件系统数据的目的。HDFS在最开始是作为Apache Nutch搜索引擎项目的基础架构而开发的。HDFS是Apache Hadoop Core项目的一部分。  HDFS有着高容错性(fault-tolerant)的特点，并且设计用来部署在低廉的(low-cost)硬件上。而且它提供高吞吐量(high throughput)来访问应用程序的数据，适合那些有着超大数据集(large data set)的应用程序。HDFS放宽了(relax)POSIX的要求(requirements)这样可以实现流的形式访问(streaming access)文件系统中的数据。 |
| Spark Core | 三个部分来解读Spark-core，首先是Spark的架构，阐述了Spark基于弹性分布式数据集RDD这个计算模型的工作机制（计算流程）：Application->Job->Stage->Task 的分解、分发和并行计算；接下去从计算模型和工作机制两个方面，分别解读RDD的设计思想及其算子，以及划分RDD有向无环图为Stage和Task、并行计算的工作机制。Spark采用了分布式计算中的Master-Slave模型。Master作为整个集群的控制器，负责整个集群的正常运行；Worker是计算节点，接受主节点命令以及进行状态汇报；Executor负责任务（Tast）的调度和执行；Client作为用户的客户端负责提交应用；Driver负责控制一个应用的执行。Spark集群启动时，需要从主节点和从节点分别启动Master进程和Worker进程，对整个集群进行控制。在一个Spark应用的执行过程中，Driver是应用的逻辑执行起点，运行Application的main函数并创建SparkContext，DAGScheduler把对Job中的RDD有向无环图根据依赖关系划分为多个Stage，每一个Stage是一个TaskSet， TaskScheduler把Task分发给Worker中的Executor；Worker启动Executor，Executor启动线程池用于执行Task。 |

# 概要设计

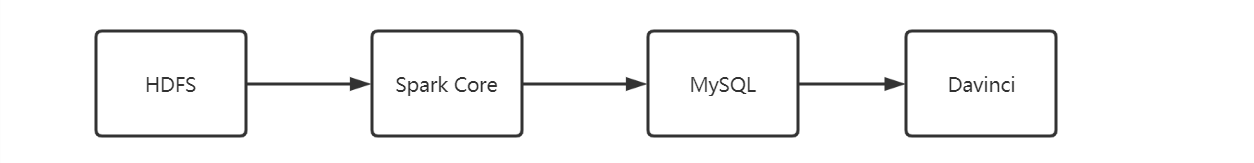
## 需求概述

对农业产品市场的统计分析由大入小，从宏观到微观，以数据为基础，具体研究领域涵盖产品类别、市场容量、产销规模、价格行情、技术特点、原材料供应、消费群体、消费结构、进出口、区域格局、品牌竞争、企业竞争、产业政策、盈利预测、市场前景等信息，准确分析了目前中国农业产品市场的发展动态、把握行业的发展方向，为各类企事业单位及其它机构经营决策提供参考依据。

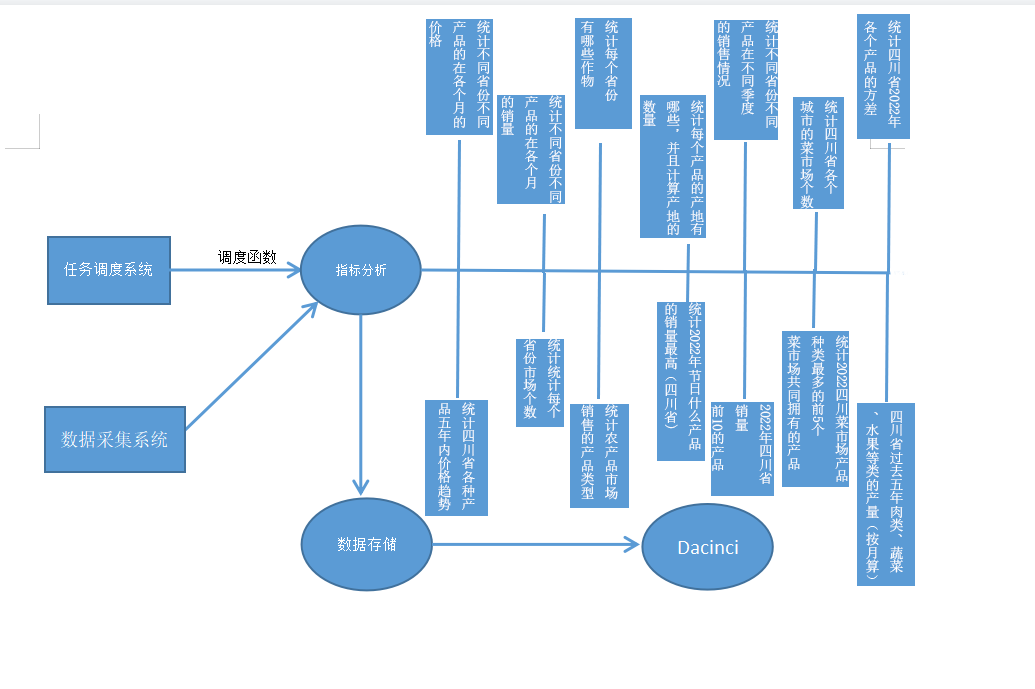
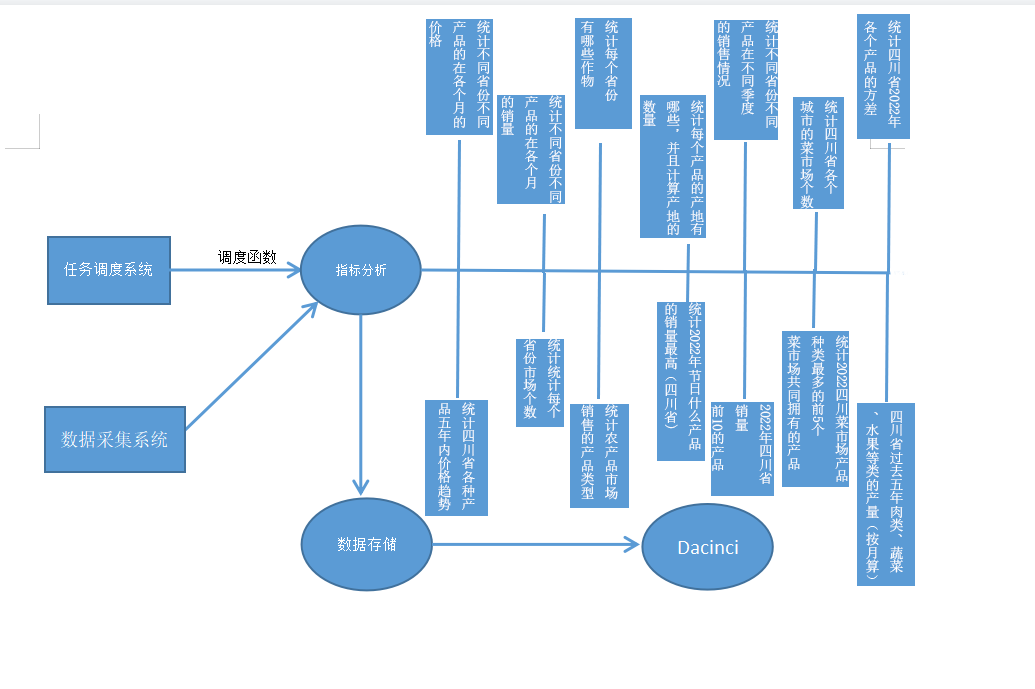
## 技术架构与选型

本项目基于Spark大数据内存计算框架，使用Scala函数式编程语言开发实现。

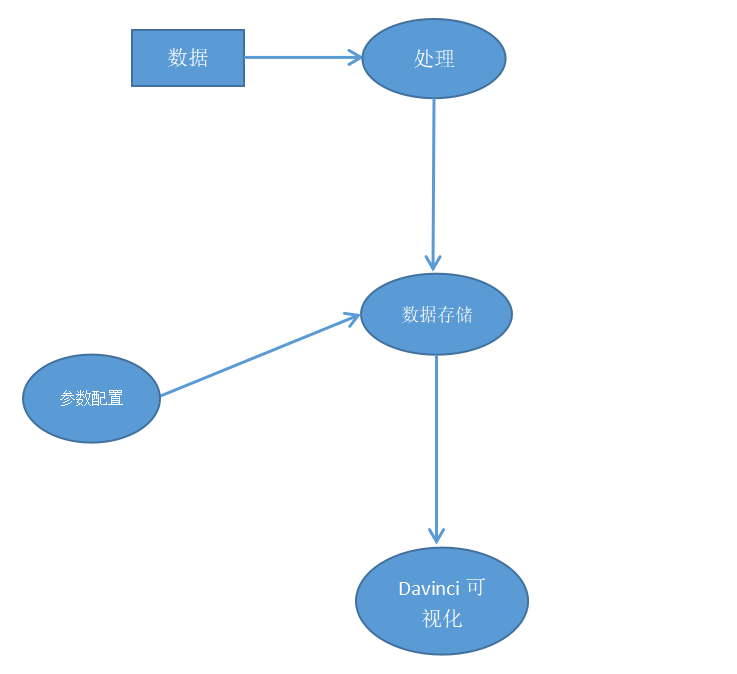
数据存放于HDFS分布式文件系统，通过Spark Core对其进行离线批处理操作，处理结果存入MySQL数据库，最后通过Davinci数据可视化工具进行结果展示。项目架构如下所示：



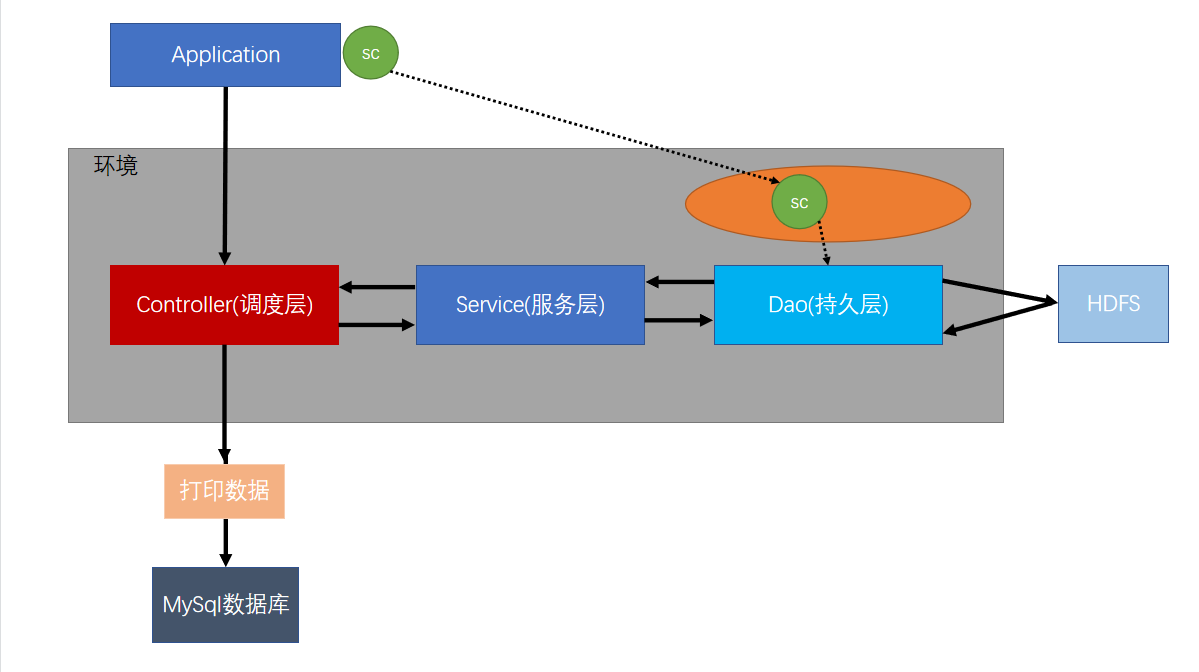
## 系统功能整体流程图



## 系统数据流图



## 项目架构流程图



# 具体实现

## 实现方法

### RDD

1. RDD是Spark中的抽象数据结构类型,Spark中最基本的数据抽象,实现了以操作本地集合的方式来操作分布式数据集的抽象实现，

2. 它代表一个不可变、可分区、里面的元素可并行计算的集合。

3. RDD Transformation算子

### 累加器

1. Driver端定义一个共享变量，将数据累加到该变量上，如果直接用foreach或map等迭代算子， 是无法将累加的变量返回到driver端，因为累加的过程发生在Executor端。一般用于计数场景下，变量往往声明在Driver端。变量在Driver端，累加的过程是在Executor端，在累加的过程Executor端是无法读取其值的，如果想读取其值，只能在Driver端才能读取。

### 广播变量

1. 将Executor端使用到的Driver端的变量生成一个副本，放到Executor端的BlockManager。有几个Executor使用到对用的变量，就有几个副本。

### 数据倾斜

1. 数据倾斜在MapReduce编程模型中十分常见,用最通俗易懂的话来说,数据倾斜无非就是大量的相同key被partition分配到一个分区里,造成了'一个人累死,其他人闲死'的情况,这种情况是我们不能接受的,这也违背了并行计算的初衷,首先一个节点要承受着巨大的压力,而其他节点计算完毕后要一直等待这个忙碌的节点,也拖累了整体的计算时间,可以说效率是十分低下的。

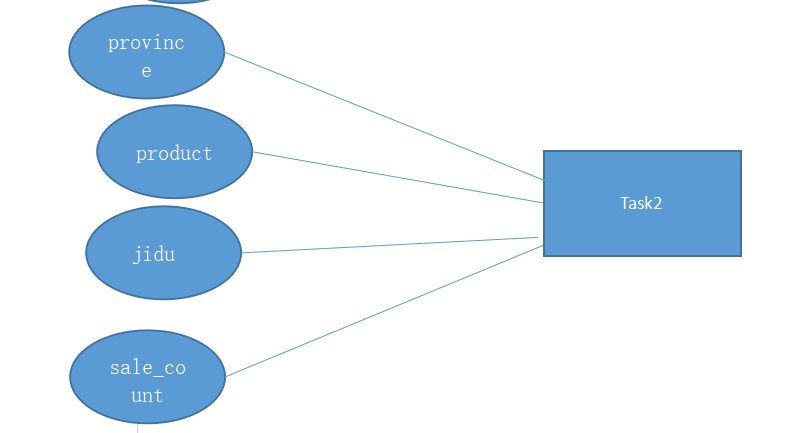
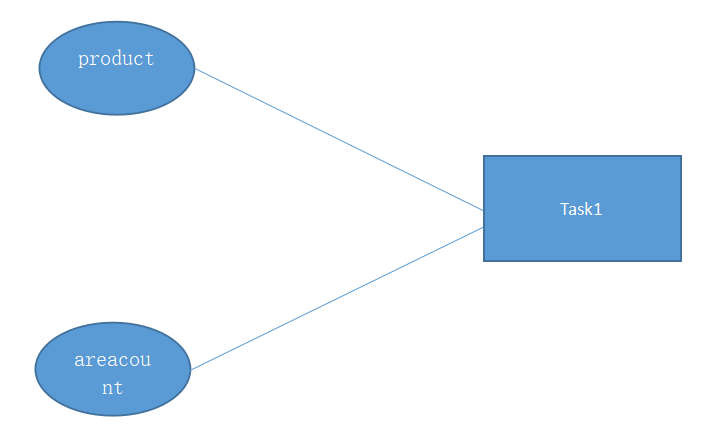
# 数据库设计

## 数据库设计综述

本系统采用MySQL数据库进行开发，本项目共有十四张表。

## 数据库逻辑结构设计

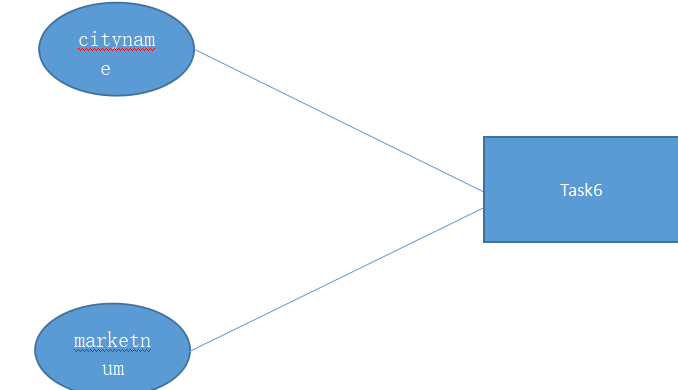
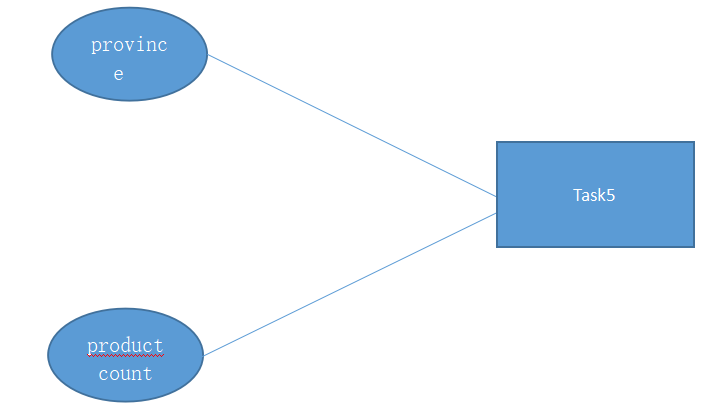
表一 表二



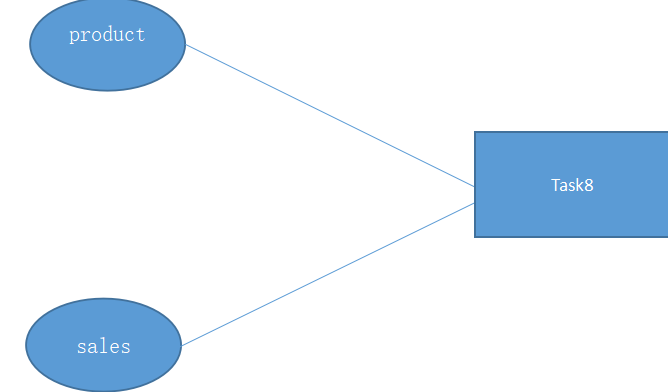
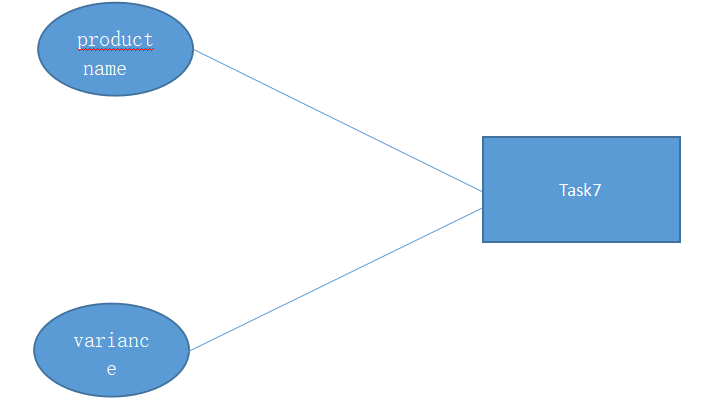
表三 表四

## 

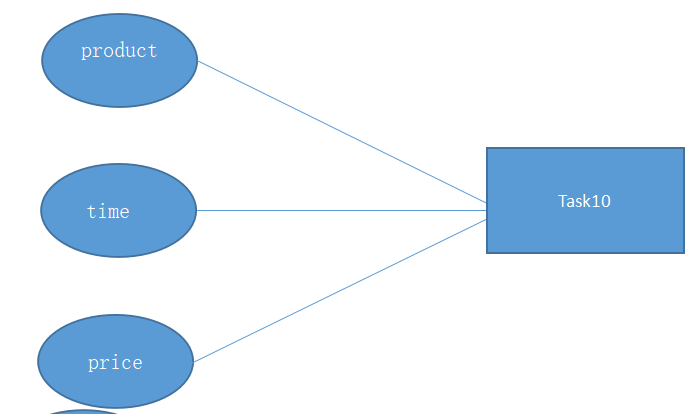
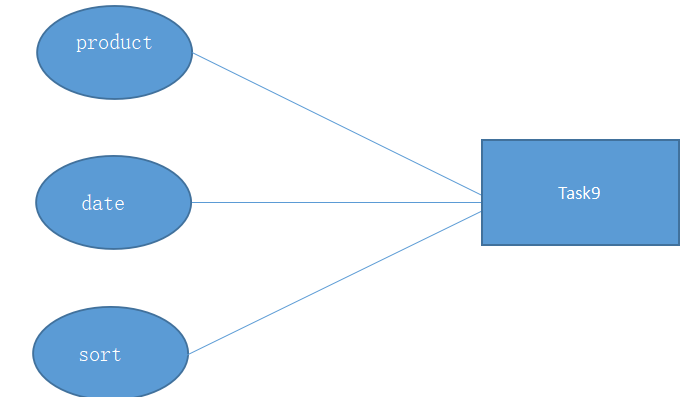
表五 表六



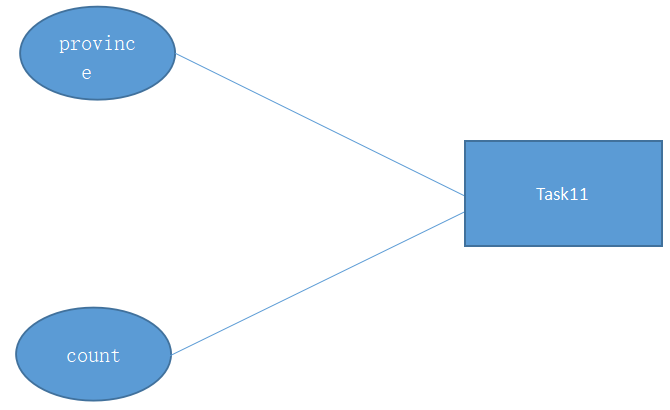
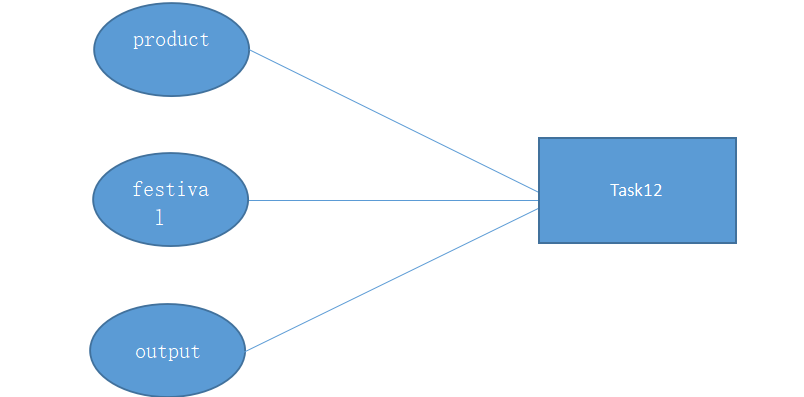
表七 表八



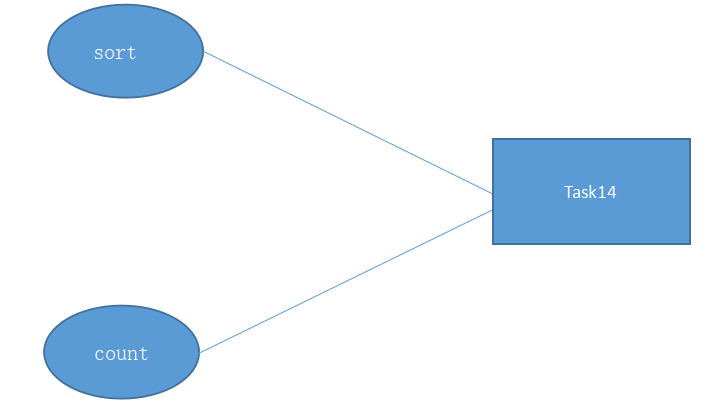
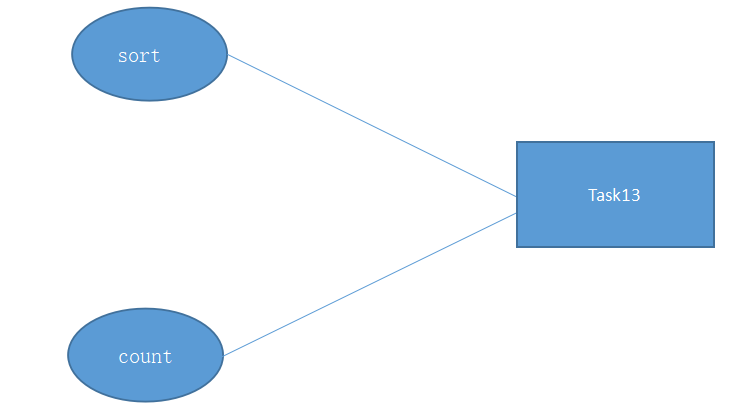
表九 表十



表十一 表十二

表十三 表十四



## 数据库物理结构设计

1.不同省份作物的价格（平均价格）论月看

def dispatch\_Print1() = {

执行需求的逻辑分析

val filterRDD = ServiveTest.dataAnalysis1() //将计算结果返回

filterRDD.foreach(println)//输出

val conn = CreateLink()

val createTabel = "CREATE TABLE task1 (product varchar(255),areacount int);"

conn.createStatement().executeUpdate(createTabel)

//TODO 把数据写到表里面去

filterRDD.foreach(

it => {

val pstat = conn.prepareStatement("insert into task1 (product,areacount) values (?,?)")

pstat.setString(1,it.\_1)

pstat.setInt(2,it.\_2)

pstat.addBatch()//addBatch()是把若干sql语句装载到一起，然后一次性传送到数据库执行，即是批量处理sql数据的

pstat.executeBatch()

pstat.close()

}

)

conn.close()

println("============已全部写入数据库============")

}

2.不同省份作物的销量（平均销量）论月看

def dispatch\_Print2() = {

//执行需求的逻辑分析

val filterRDD = ServiveTest.dataAnalysis2() //将计算结果返回

filterRDD.foreach(println)//输出

val conn = CreateLink()

val createTabel = "CREATE TABLE task2 (province varchar(255),product varchar(255),jidu varchar(255),sale\_count int);"

conn.createStatement().executeUpdate(createTabel)

//TODO 把数据写到表里面去

filterRDD.foreach(

it => {

val pstat = conn.prepareStatement("insert into task2 (province,product,jidu,sale\_count) values (?,?,?,?)")

pstat.setString(1,it.\_1)

pstat.setString(2,it.\_2)

pstat.setString(3,it.\_3)

pstat.setInt(4,it.\_4)

pstat.addBatch()//addBatch()是把若干sql语句装载到一起，然后一次性传送到数据库执行，即是批量处理sql数据的

pstat.executeBatch()

pstat.close()

}

)

conn.close()

println("============已全部写入数据库============")

}

1. 每个省份作物种类
2. 产品的产地情况，看看哪个产品的产地最少
3. 某一年不同产品在不同地区的产量情况
4. 不同省份2021年不同季度粮食销售情况
5. 统计四川省各城市的市场个数
6. 节日(2021春节、冬至、端午、中秋)什么产品的销量最高（四川省)·节日前几天销量 涨幅最大的几种产品的销量统计
7. 某种产品五年内价格趋势(按月分析)
8. 2021年四川省销量前10的产品
9. 四川省过去五年肉类、蔬菜、水果等类的产量
10. 四川省2021年各个产品的价格波动情况（方差）

（以上处理方法一致）

1. 统计农产品市场销售的产品类型·使用数据倾斜

def dispatch\_Print13() = {

val resultRDD = ServiveTest.dataAnalysis13()

resultRDD.foreach(println)

val conn = CreateLink()

val sql = "CREATE TABLE task13\n(\nproductType varchar(10),\nnum Int);"

val stmt = conn.createStatement()

stmt.executeUpdate(sql)

resultRDD.foreach{

action => {

val pstat = conn.prepareStatement("insert into task13 (productType ,num) values (?,?)")

pstat.setString(1,action.\_1)

pstat.setInt(2,action.\_2)

pstat.addBatch()

pstat.executeBatch

pstat.close()

}

}

conn.close()

println("============已全部写入数据库============")

}

1. 2021四川菜市场产品种类最多的前5个菜市场共同拥有的产品·使用广播变量

def dispatch\_Print14() = {

val resultRDD = ServiveTest.dataAnalysis14()

resultRDD.foreach(println)

val conn = CreateLink()

val sql = "CREATE TABLE task14\n(\n产品 varchar(10));"

val stmt = conn.createStatement()

stmt.executeUpdate(sql)

resultRDD.foreach{

action => {

val pstat = conn.prepareStatement("insert into task14 values (?)")

pstat.setString(1,action)

pstat.addBatch()

pstat.executeBatch

pstat.close()

}

}

conn.close()

println("============已全部写入数据库============")

}

## 数据字典

数据结构：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **描述** | **是否可为空** | **备注** |
| **name** | String | 农产品名称 | NOT NULL |  |
| **price** | Double | 农产品价格 | NOT NULL |  |
| **crawl\_time** | TimeStamp | 采集时间 | NOT NULL |  |
| **market** | String | 农产品市场名称 | NOT NULL |  |
| **province** | String | 省份 | NOT NULL |  |
| **city** | String | 城市 | NOT NULL |  |

# 项目实用性分析

农贸市场不仅是承载了一代代人日常生活的市井喧嚣之地，同时也是一个地区变迁的记录器，随着一个个新居住社群的建立，会有相应的配套市场应运而生。一个个农贸市场的出现和消失，可以归纳出城市发展的方向和重点区域的转移，也记录着人们生活过的地方，以及生活的气息，和农贸市场一同离开，远去。

通过对数据模型的分析，我们可以从多个角度出发，研究数据背后的规律。而这些规律，又可以被各方所用，促进市场的良性发展。数据容量越大，涵盖面越广，越利于我们分析出其背后蕴含的规律。针对全国各地的收货商，我们可以为其提供不同省份的农作物价格变化趋势、各个省份作物分布情况等有效数据，帮助其更好地规划未来的收货计划。针对四川省的从业人员，我们可以详细地给出整体的市场规模，全年的价格波动情况，以及特殊节日农产品销售情况等有效数据，方便相关人员及时对市场进行调整。